Adaptive Robust Extended Kalman Filter

Adaptive Robust Extended Kalman Filter for GNSS-Based Orbit Determination

Francisco Monteiro Esteves Nunes

Thesis to obtain the Master of Science Degree in

Integrated Master Degree (MSc) in Aerospace Engineering

Supervisor: Hugo José Dias Lopes
Co-Supervisor: Fernando Nunes

Examination Committee
Chairperson: João Miranda Lemos
Supervisor: Hugo José Dias Lopes
Members of the Committee: José Sanguino

October 2015
You are always a student, never a master.
You have to keep moving forward.

Conrad Hall
Acknowledgments

People often say that education is what replaces an empty mind with an open one. I was fortunate enough to learn from the most motivated, dedicated and intelligent people. Since my early teachers to my latest Professors.

I would like to thank my Supervisor and Engineer Hugo Lopes for his incessant motivation and support throughout this thesis. Also my Co-Supervisor, Prof. Fernando Nunes for his valuable inputs. Also to Instituto Superior Técnico and Deimos Engenharia for making this possible.

To my family for their unconditional support and comprehension. Their belief makes me halfway there.

A special thanks to my friends, from the early P, R and J, to the very special (double T’s and R) V’s. To M, N, P & A C, not forgetting R. My colleagues turned friends, beginning with JE, LG and FC, to the latest discoveries of M’s, R and I.
Abstract

A GNSS-based space receiver is able to determine autonomously the spacecraft state in real-time. The Extended Kalman Filter (EKF) is one of the most used methods for state estimation in aerospace applications due to its simplicity, reliability and flexibility, although, in order for the EKF to guarantee a satisfactory performance, the system model should be known exactly. Unknown external disturbances may lead to inaccuracy of the state estimate or even divergence. Also, computational effort must be taken into account since space receivers have a very limited computational capacity.

In this thesis, the development of Kalman filters designed for state estimation of the position and velocity for external accelerations beyond current implementations is attempted and their performance evaluated. Three Kalman Filters are developed, each with its unique characteristics: the Extended Kalman Filter, the Robust Extended Kalman Filter and the Adaptive Robust Extended Kalman Filter. The three filters are implemented assuming the same system model, for a proper comparison analysis.

Regarding the simulation, three different scenarios are modeled with different sources of non-modeled accelerations. A Low Earth Orbit with a perturbations model, an orbit with outliers and trajectory correction maneuvers. Results show that it is possible to reduce the error of the estimation when non-modeled accelerations are present. The best error performance based filter is the Adaptive Robust Extended Kalman Filter.

Keywords

Adaptive Robust Extended Kalman Filter (AREKF), Robust Extended Kalman Filter (REKF), Extended Kalman Filter (EKF), Acceleration Model Mismatch, Orbit Generation, Noise modeling, AREKF performance comparison, α parameter analysis
Resumo

Um receptor de GNSS para aplicações espaciais consegue determinar autonomamente e em tempo real o estado da nave. O Extended Kalman Filter (EKF) é um dos métodos mais utilizados na estimação de estados em aplicações aeroespaciais, devido à sua simplicidade, fiabilidade e adaptabilidade, no entanto, de modo a ao EKF garantir um desempenho satisfatório é necessário conhecer exactamente o modelo do sistema. Perturbações desconhecidas externas ao sistema podem levar a uma degradação da performance do filtro e, em último caso, à sua divergência. Em aplicações espaciais o poder de computação é também um factor relevante, dada a limitada capacidade de computação dos receptores.

Ao longo de trabalho é desenvolvido um conjunto de filtros de Kalman com o objectivo de estimar a posição e velocidade sujeitos a acelerações externas. São implementados métodos que vão além do comercialmente feito actualmente e é feito uma comparação de performance. São desenvolvidos três filtros de Kalman: o Extended Kalman Filter, o Robust Extended Kalman Filter e o Adaptive Robust Extended Kalman Filter. Para uma comparação correcta, nos três filtros é assumido o mesmo modelo de sistema.

Estes três filtros são testados em três ambientes diferentes: uma órbita baixa (LEO) calculada com um modelo de perturbações, uma órbita com erros pontuais (outliers) nas medições e uma manobra de correção de trajectória. Os resultados evidenciam que é possível diminuir o erro de estimação dos métodos actualmente implementados. Numa análise de performance de erro o filtro com o melhor desempenho é o Adaptive Robust Extended Kalman Filter.

Palavras Chave

Adaptive Robust Extended Kalman Filter (AREKF), Robust Extended Kalman Filter (REKF), Extended Kalman Filter (EKF), Modelação de Acelerações Desconhecidas, Geração de Órbitas, Modelação de Ruído, Comparação de performance de AREKF, análise de parâmetros
## Contents

1 Introduction ........................................... 1
   1.1 Thesis Background .................................... 2
   1.2 Research Objectives .................................. 3
   1.3 Thesis Outline ....................................... 4

2 Fundamentals ............................................ 5
   2.1 Filtering .............................................. 6
      2.1.1 Nonlinear filtering .................................. 6
      2.1.2 Kalman Filtering .................................... 7
         2.1.2.A Discrete Kalman Filter ............................. 7
         2.1.2.B Kalman Filter Dynamics ............................ 8
      2.1.3 Extended Kalman Filter ............................... 10
      2.1.4 Robust Extended Kalman Filter ....................... 11
         2.1.4.A Stability Analysis ................................. 13
      2.1.5 Adaptive and Robust Extended Kalman Filter .......... 15
         2.1.5.A Stability Analysis ................................. 15
   2.2 Reference Frames ...................................... 18
      2.2.1 Earth-Centered Inertial Coordinate System .......... 18
      2.2.2 Earth-Centered Earth-Fixed Coordinate System .... 18
   2.3 Fundamentals of Satellite Navigation ..................... 19
      2.3.1 Orbital Mechanics .................................... 19
         2.3.1.A Orbital Design ................................. 23
         2.3.1.B Simplified Perturbations Models .................. 23

3 Modeling and Implementation .............................. 25
   3.1 Orbit Generation ...................................... 26
      3.1.1 GPS Constellation Generation ....................... 26
      3.1.2 Reference Orbits Generation ......................... 26
   3.2 Filter Implementation ................................... 29
      3.2.1 Equations of Motion ................................ 29
      3.2.2 Dynamic Model ..................................... 31
         3.2.2.A Receiver’s clock state space ..................... 32
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The three Keplerian orbital elements defining the shape of the satellite’s orbit</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>The three Keplerian orbital elements defining the orientation of the orbit</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>GPS Constellation Satellites in ECEF coordinates</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>Reference LEO orbit simulated in ECEF coordinates</td>
<td>27</td>
</tr>
<tr>
<td>3.3</td>
<td>Reference orbit used as maneuvers example</td>
<td>28</td>
</tr>
<tr>
<td>3.4</td>
<td>PV model state space</td>
<td>31</td>
</tr>
<tr>
<td>3.5</td>
<td>Receiver clock state space</td>
<td>32</td>
</tr>
<tr>
<td>3.6</td>
<td>Evolution of $\sigma_{D\text{LL}}$ with a linear $C/N_0$</td>
<td>35</td>
</tr>
<tr>
<td>3.7</td>
<td>Receiver antenna pattern characterization</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Reference orbit with modeled noise for a LEO orbit</td>
<td>40</td>
</tr>
<tr>
<td>4.2</td>
<td>GPS Constellation Satellites $C/N_0$ for one orbital period of a LEO orbit</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>Availability of GPS constellation for one orbital period of a LEO orbit</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>EKF error for $q_v = 2 \ (m/s)^2$</td>
<td>43</td>
</tr>
<tr>
<td>4.5</td>
<td>EKF error for $q_v = 0.001 \ (m/s)^2$</td>
<td>44</td>
</tr>
<tr>
<td>4.6</td>
<td>REKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $q_v = 2 \ (m/s)^2$</td>
<td>45</td>
</tr>
<tr>
<td>4.7</td>
<td>REKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $q_v = 0.001 \ (m/s)^2$</td>
<td>45</td>
</tr>
<tr>
<td>4.8</td>
<td>REKF RMSE variation with $\gamma$ parameter and $q_v$ for REKF</td>
<td>46</td>
</tr>
<tr>
<td>4.9</td>
<td>AREKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $\alpha = 0.6$ and $q_v = 2 \ (m/s)^2$</td>
<td>48</td>
</tr>
<tr>
<td>4.10</td>
<td>AREKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $\alpha = 0.6$ and $q_v = 0.001 \ (m/s)^2$</td>
<td>48</td>
</tr>
<tr>
<td>4.11</td>
<td>AREKF RMSE evolution with $\alpha$, for $\gamma = 1.00005\max(eig(P))$ and $q_v = 0.001 \ (m/s)^2$</td>
<td>49</td>
</tr>
<tr>
<td>4.12</td>
<td>All three algorithms error norm evolution, for $\gamma = 3\sqrt{\max(eig(P))}$, $\alpha = 0.7$ and $q_v = 0.001 \ (m/s)^2$</td>
<td>50</td>
</tr>
<tr>
<td>4.13</td>
<td>All three algorithms RMSE evolution with $\alpha$, for $\gamma = 3\sqrt{\max(eig(P))}$ and $q_v = 0.001 \ (m/s)^2$</td>
<td>51</td>
</tr>
<tr>
<td>4.14</td>
<td>The three algorithms RMSE with outliers for $t = 150s$ and $t = 151s$</td>
<td>52</td>
</tr>
<tr>
<td>4.15</td>
<td>EKF error for orbital maneuvers</td>
<td>53</td>
</tr>
<tr>
<td>4.16</td>
<td>All three algorithms error norm with orbital maneuvers</td>
<td>53</td>
</tr>
<tr>
<td>B.1</td>
<td>Innovations Example</td>
<td>B-2</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Kalman Filter variables ......................................................... 8
2.2 GPS Ephemeris data definitions ............................................. 22
2.3 Computation of a Satellite’s ECEF Position Vector ([Kaplan, 2005]) ..................... 22

3.1 Parameters for the Allan variance of several oscillators ([Winkel, 2003]) .......................... 33
3.2 Typical error for pseudoranges for GPS L1 C/A ........................................... 37

4.1 Environment Characteristics ..................................................... 42
4.2 EKF RMSE results for $q_v = 2 \text{(m/s)}^2$ and $q_v = 0.001 \text{(m/s)}^2$ for 20 runs .......... 44
4.3 REKF RMSE results for $q_v = 2 \text{(m/s)}^2$ for 20 runs ...................................... 46
4.4 REKF RMSE results for $q_v = 0.001 \text{(m/s)}^2$ for 20 runs .................................. 46
4.5 AREKF RMSE results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 2 \text{(m/s)}^2$ for 20 runs 49
4.6 AREKF RMSE results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 0.001 \text{(m/s)}^2$ for 20 runs 49
4.7 Computational times for $\gamma = 3 \sqrt{\text{max}(\text{eig}(P))}$, $\alpha = 0.7$ and $q_v = 2 \text{(m/s)}^2$ for 30 runs on a MacOS 2.4 GHz Intel Core 2 Duo™ with 10 GB 1067 MHz DDR3 .................................. 51
4.8 All three algorithms RMSE with outliers for $t = 150$ and $t = 151$ ............................ 52
4.9 All three algorithms RMSE with orbital maneuvers .............................................. 54

B.1 AREKF RMSE complete results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 0.001 \text{(m/s)}^2$ for 20 runs .................................................. B-3
Abbreviations

**AREKF**  Adaptive Robust Extended Kalman Filter

**ECEF**  Earth-Centered Earth Fixed

**ECI**  Earth-Centered Inertial

**EKF**  Extended Kalman Filter

**GEO**  Geostationary Earth Orbit

**GNSS**  Global Navigation Satellite System

**GPS**  Global Positioning System

**HEO**  Highly Elliptical Orbit

**MEO**  Medium Earth orbit

**KF**  Kalman Filter

**LEO**  Low-Earth Orbit

**pdf**  Probability Density Function

**REKF**  Robust Extended Kalman Filter

**RMSE**  Root Mean Square Error

**SDCM**  Square Root of the Trace (Diagonal) of the Covariance Matrix

**SEM**  System Effectiveness Model

**SGP**  Simplified Perturbations Models

**TLE**  Two-Line Elements
Introduction

Contents

1.1 Thesis Background .................................................. 2
1.2 Research Objectives .................................................. 3
1.3 Thesis Outline ......................................................... 4
In this first chapter, the thesis subject is presented. The thesis motivation and general problem are first presented, in Section 1.1. Then the state of the art and background are presented in Section 1.2. Lastly, the structure of the thesis can be consulted in Section 1.3.

1.1 Thesis Background

The Global Navigation Satellite System (GNSS) based space navigation with Kalman filters is widely used for positioning and velocity determination applications.

Determining orbits of celestial bodies started many years ago with Kepler (1610) and Legendre (1750) making the first efforts. Gauss (1810) gave it a firm analytical and computational basis. Many innovations took place between the original foundations and current theories, but the fundamental principles are the same. Swerling (1959) and Kalman (1960) gave it a big effort with a recursive method, which retains Kalman’s name. Although the recursive formulations of Swerling and Kalman are very popular today, they are only a variation of the fundamental contributions of Gauss and Legendre.

Satellite orbit determination can be described as the method of determining the position and velocity (i.e., the state/ephemeris) of an orbiting object such as an interplanetary spacecraft or an Earth orbiting satellite. The object’s motion is approximated by a set of equations with the state adjusted in response to a set of discrete observations and subject to both random and systematic errors. The spacecraft is usually assumed to be influenced by a variety of external forces, including gravity, atmospheric drag, solar radiation pressure, third-body perturbations, Earth tidal effects, and general relativity in addition to satellite propulsive maneuvers. The complex description of these forces results in a highly nonlinear set of dynamical equations of motion. Furthermore, the lack of detailed knowledge of the physics of the environment through which the satellite travels, limits the accuracy with which the states of the satellite can be determined at any given time. Similarly, observational data are inherently nonlinear with respect to the state of the spacecraft. Since the orbit determination equations are also highly nonlinear, linearization is usually performed so that linear estimation techniques can be used to resolve the orbit determination problem.

A GNSS-based space receiver is able to determine autonomously the spacecraft states in real-time. The typical navigation solution is composed by the spacecraft position, velocity and modeling parameters of the receiver clock behavior. The satellite positioning using Global Navigation Satellite System is typically based on Kalman filtering schemes. The main advantages of the Kalman filter approach with respect to other algorithms can be summarized in the following:

1. A statistical filter increases the accuracy of the solution in presence of random perturbations;

2. The implemented dynamic model makes the filter able to predict position and velocity at future time, even in absence of good measurements, overcoming the discontinuity in the Standard Position Solution due to loss of Global Navigation Satellite System satellite visibility.

The common Extended Kalman Filter (EKF) algorithm has been implemented and validated in the last decade in spacecraft navigation. Its results have been proven to be very acceptable for nominal
orbital trajectories. However, the Extended Kalman Filter is not reliable in a situation of external disturbances, such as thrusters, for instance during an orbit transfer or a manoeuvre. This external disturbance can result in filter divergence if not accounted for or require sub-optimal filter tuning to ensure correct operation during worst-case scenarios. Recent advances in the filtering topic can be applied to avoid these problems, such as the Adaptive Robust Extended Kalman Filter (AREKF).

This work provides an estimator that enhances the ability of the Extended Kalman Filter to handle large external disturbances (abundant in orbital maneuvers) and observation errors (outliers). For both model-plant mismatch and disturbances, other methods are proposed, some more effective than others. One of the most famous is the extended robust $H_{\infty}$ filter where the error covariance matrix is designed in order to accommodate more terms that scale the magnitude of the model error.

A robust adaptive Kalman filter for linear systems using stochastic uncertainties was proposed in (Wang, 1999). Some years before, (Julier, 1997) proposed an algorithm for highly nonlinear state and observation models that uses a deterministic sampling technique that choses a set of points (sigma points) around the mean and result in a more accurately result for the mean and covariance.

Other methods were also first taken into account such as the Gaussian Sum Approximations Filter (Alspach & Sorenson, 1972) and the Interacting Multiple Model (Blom, 1988). The Gaussian Sum Approximations method was first ruled out, since it lacks the adaptive feature of the state estimation. The Interacting Multiple Model using Kalman Filters, frequently used in target tracking, is a possible and a priori also viable method, but is left out of this comparison. Also, particle filters were not taken into consideration due to the high computational requirements these methods demand.

1.2 Research Objectives

The objective of this work is to investigate and compare three algorithms - the Extended Kalman Filter, the Robust Extended Kalman Filter and the Adaptive Robust EKF - applied to the position and velocity estimation of a spacecraft carrying a GNSS-based space receiver in situations of model mismatch.

A robust, adaptive, nonlinear, low-computational power and easy to implement algorithm are the characteristics desired for the algorithm.

Algorithms that autonomously determine the spacecraft state in real-time with a low computational power is of most importance for space applications and is an open field discussed in this thesis. The theoretical approach for the three methods is made, with the AREKF being more detailed than the other two methods since the AREKF is a fairly new method and still not applied in real life situations. This novel approach may outcome in performance the existing algorithms commercially used.
1.3 Thesis Outline

This document consists of five chapters and two appendices which are arranged in the following order.

The first chapter introduces the thesis background and research objectives.

The second chapter, Section 2.1, covers the background theory used in this work. A brief review of the Kalman Filtering and one of its non-linear adaptations - Extended Kalman Filter - is made in Sections 2.1.2 and 2.1.3, respectively. Section 2.1.4 covers the Robust Extended Kalman Filter (REKF) and finally the Adaptive Robust Extended Kalman Filter is presented in Section 2.1.5. Stability analysis is also performed, on both the REKF and AREKF sections. This chapter ends with section 2.2, where the reference frames used - Earth-Centered Inertial (ECI) and Earth-Centered Earth Fixed (ECEF) - systems are described. On Section 2.3 the fundamentals of Satellite Navigation are presented and the simplified perturbations model is briefly discussed.

The third chapter, Section 3, discusses implementation issues. On Section 3.1 the orbit generation method is briefly described and Section 3.2 presents the filter implementation issues, where the dynamic and observation model are derived. It is also explained how the noise was modeled, on Section 3.2.4, and how to properly tune the Robust Extended Kalman Filter and Adaptive Robust Extended Kalman Filter parameters.

Fourth chapter, Section 4, discusses the results of the simulation. The simulation characteristics of the reference orbit used, a description of the $C/N_0$ and other characteristics are presented on Section 4.1. The Simulation Results are covered in Section 4.2. First, the EKF results for two values of parameter $q_v$ are presented and discussed in Section 4.2.1, followed by the REKF - Section 4.2.2 - results, where an analysis on $\gamma$ parameter is made. Next, the AREKF and the parameter $\alpha$ are studied in Section 4.2.3, with special attention given to the filter output for the variation of $\alpha$.

In Section 4.2.4 a comparison between algorithms is made for the Low-Earth Orbit (LEO) orbit, and then in Section 4.2.5 the comparison is made for other two environments: measurement outlier - 4.2.5.A - and orbital maneuvers - 4.2.5.B.

The fifth chapter contains final remarks of this thesis and recommendations for future work.

The appendices contain the proof of the Stability Theorems in Appendix A and some more complete results of the Results section, in Appendix B.
Contents

2.1 Filtering ......................................................... 6
2.2 Reference Frames ............................................... 18
2.3 Fundamentals of Satellite Navigation .............................. 19

Fundamentals
The first section intends to cover the main features of the filtering techniques used in this work. A brief review of the Kalman Filtering and one of its non-linear adaptations - Extended Kalman Filter - is made in Sections 2.1.2 and 2.1.3 respectively. Section 2.1.4 covers the Robust Extended Kalman Filter and finally the Adaptive Robust Extended Kalman Filter is presented in Section 2.1.5.

Section 2.2 covers the reference frames used - ECI, ECEF - systems. On Section 2.3 the fundamentals of Satellite Navigation are presented and the simplified perturbations model is briefly described.

2.1 Filtering

2.1.1 Nonlinear filtering

Consider that the process to be estimated is the random vector sequence $x_1, x_2, \ldots$ described by the difference equation

$$x_{k+1} = \Phi(x_k) + \Gamma_k u_k, \quad k = 1, 2, \ldots$$ (2.1)

$$x_1 = C$$ (2.2)

where $\Phi(\cdot) : IR^n \to IR^n$ is a nonlinear vector function, $\Gamma_k$ is a $n \times s$ matrix, $u_1, u_2, \ldots$ is a white zero-mean Gaussian vector sequence with covariance matrix $Q$, and $C$ is a vector random variable whose Probability Density Function (pdf) $p(x_1)$ characterizes the a priori knowledge about $x_1$.

The observations are produced by the equation

$$z_k = h(x_k) + v_k, \quad k = 1, 2, \ldots$$ (2.3)

where $h(\cdot) : IR^n \to IR^m$ is a nonlinear vector function and $v_1, v_2, \ldots$ is a white zero-mean Gaussian vector sequence with covariance $R$. This sequence is independent of the sequence $u_1, u_2, \ldots$ and the random variable $x_1$.

Let $Z_k = \{z_1, z_2, \ldots, z_k\}$ be the set of observations from the initial time up to the current time $k$. The nonlinear filtering problem consists of propagating the conditional pdf $p(x_k|Z_k)$, also called a posteriori or filtering density. The theoretical solution applies iteratively the Bayes law and the Chapman-Kolmogorov equation for discrete Markov processes (Jazwinski, 1970). Each iteration is carried out in two steps [Leitao, 1983]

Prediction:

$$p(x_{k+1}|Z_k) = \int_{IR^n} p(x_{k+1}|x_k) p(x_k|Z_k) dx_k$$ (2.4)

Filtering:

$$p(x_k|Z_k) = c_k p(z_k|x_k) p(x_k|Z_{k-1})$$ (2.5)

where

$$c_k = \left[ \int_{IR^n} p(z_k|x_k)p(x_k|Z_{k-1}) dx_k \right]^{-1}$$ (2.6)
is a normalization factor. The quantities \( p(x_k|Z_{k-1}) \) and \( p(x_k|Z_k) \) are called respectively prediction and filtering densities of iteration \( k \).

After the filtering density \( p(x_k|Z_k) \) is obtained, the optimal estimate \( \hat{x}_{k,\text{opt}} \) is determined by minimizing the conditional mean of a pre-defined cost function \( L(\hat{x}_k - x_k) \)

\[
\int_{\mathbb{R}^n} L(\hat{x}_k - x_k) p(x_k|Z_k) \, dx_k \xrightarrow{\text{minimization}} \hat{x}_{k,\text{opt}}
\]  

(2.7)

The implementation of the optimal nonlinear filter is unfeasible due to the convolution operation \((2.4)\) which requires, in general, a huge computational effort. Besides, the filter and prediction pdfs are, in general, defined by an infinite number of parameters. Therefore, one resorts to other suboptimal algorithms, including the EKF. Note also that, if the dynamics function \( \Phi \) and the observations function \( h \) depend linearly on \( x_k \) and \( x_1 \) has a Gaussian distribution, the operations of prediction and filtering produce only Gaussian functions \( p(x_k|Z_{k-1}) \) and \( p(x_k|Z_k) \) and the propagation equations are very simple to compute. In that case, the filter updating consists of determining the means and the covariances of the prediction and filtering densities and the nonlinear filtering problem is converted into a linear problem whose (exact) solution is the (linear) Kalman filter.

2.1.2 Kalman Filtering

The Kalman filtering concept was first developed by Rudolph Kalman in 1960 and consists of a linear, discrete-time, time-varying system characterized by a finite dimensional state vector and a sequence of noisy observations that allow to evaluate the state estimate through minimizing the mean-square error \( [\text{Kalman, 1960}] \). To achieve its goal, consecutive cycles of prediction and filtering are needed, which yield the filter dynamics. The dynamics of these cycles may be derived and interpreted in the framework of Gaussian pdf \( [\text{Jazwinski, 1970}] \).

The Kalman Filter (KF) is intended to be an effective and versatile mathematical procedure in which noisy sensor outputs are combined to estimate the state of a system with uncertain dynamics \( [\text{Grewal et al., 2001}] \).

In Chapter 3, the parameters used by the filter are identified as well as the expected output.

2.1.2.A Discrete Kalman Filter

Consider a linear, time-varying system with the following dynamics (discrete-time):

\[
x_{k+1} = A_k x_k + B_k u_k + G_k w_k \quad t \geq 0
\]  

(2.8)

\[
y_k = H_k x_k + v_k
\]  

(2.9)

where the variables in Eq. (2.8) and Eq. (2.9) represent:
Table 2.1: Kalman Filter variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_k$</td>
<td>State vector</td>
<td>$[n \times 1]$</td>
</tr>
<tr>
<td>$u_k$</td>
<td>Control input vector</td>
<td>$[l \times 1]$</td>
</tr>
<tr>
<td>$y_k$</td>
<td>Measurements vector</td>
<td>$[p \times 1]$</td>
</tr>
<tr>
<td>$w_k$</td>
<td>System/Process noise vector</td>
<td>$[m \times 1]$</td>
</tr>
<tr>
<td>$v_k$</td>
<td>Measurements noise vector</td>
<td>$[p \times 1]$</td>
</tr>
<tr>
<td>$A_k$</td>
<td>System (transition) matrix</td>
<td>$[n \times n]$</td>
</tr>
<tr>
<td>$B_k$</td>
<td>Input (distribution) matrix</td>
<td>$[n \times l]$</td>
</tr>
<tr>
<td>$G_k$</td>
<td>System noise input matrix</td>
<td>$[n \times m]$</td>
</tr>
<tr>
<td>$H_k$</td>
<td>Observation matrix</td>
<td>$[p \times n]$</td>
</tr>
</tbody>
</table>

The noise processes $w_k$ and $v_k$ are sequences of white, zero mean, Gaussian noise and have covariance matrices $Q_k$ and $R_k$ respectively, both positive definite (Simon, 2006):

$$w_k \sim (0, Q_k)$$
$$v_k \sim (0, R_k)$$

$$E[w_k w_j^T] = Q_k \delta_{kj}$$
$$E[v_k v_j^T] = R_k \delta_{kj}$$
$$E[v_k w_j^T] = 0 \ \forall \ k, j$$

(2.10)

### 2.1.2.2 Kalman Filter Dynamics

The issue of the state estimation can be considered as the estimation of a random parameter vector $(x_k)$. On the above system Eq. 2.8 and Eq. 2.9, the KF is the filter that provides the minimum mean-square state error estimate. If the initial state is a Gaussian vector, the dynamics and observations are linear and the dynamics and observations noises are white and Gaussian:

1. The conditional probability density function $p((x_k)|Y_1^k, U_0^{k-1})$ are Gaussian for any $k$;
2. The KF propagates the conditional pdf $p[x_k|Y_1^k, U_0^{k-1}]$ and obtains the state estimate. This is possible through optimizing a given criteria. The KF is the best filter among all the possible filter types and it optimizes any criteria that might be considered (Ribeiro, 2004).

Assuming that:

$$p((x_k)|Y_1^k, U_0^{k-1}) \sim N(\hat{x}_k|k, P_k|k)$$

(2.11)

represents a Gaussian pdf, the state estimate $\hat{x}_k|k$ is the mean of the pdf. The covariance matrix $P(k|k)$ expresses the uncertainty of the state’s estimate.

$$\hat{x}_k|k = E[x_k|Y_1^k, U_0^{k-1}]$$

$$P_k|k = E[(x(k) - \hat{x}_k|k)(x_k - \hat{x}_k|k)^T|Y_1^k, U_0^{k-1}]$$

(2.12)

which means that rather than propagating the entire pdf as in the nonlinear filtering problems, the KF only propagates the first (mean) and second (covariance) moments because the propagator pdfs

---

1. Where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$

2. $U_0^{k-1} = u_0, u_1, ..., u_{k-1}$ and $Y_1^k = y_1, y_2, ..., y_k$
are gaussian. This propagation is recursive: to evaluate $\hat{x}_{k+1|k+1}$, the filter only requires the previous estimate, $\hat{x}_{k|k}$ and the new observation $y_{k+1}$.

The estimation consists of two steps. The Time Update (Prediction) step, where “$k + 1|k$” stands for a prediction estimate and the Measurement Update (Correction) step, where “$k + 1|k + 1$” describes a correction estimate.

On the prediction step, a deviation between the estimated state ($\hat{x}$) and the true state ($x$) as well as an error covariance matrix are defined:

$$\tilde{x}_{k+1|k} := x_{k+1} - \hat{x}_{k+1|k}$$
$$P_{k+1|k} = E[\tilde{x}_{k+1|k} \tilde{x}_{k+1|k}^T]$$  \hfill (2.13)

The same is defined for the correction step:

$$\tilde{x}_{k+1|k+1} := x_{k+1} - \hat{x}_{k+1|k+1}$$
$$P_{k+1|k+1} = E[\tilde{x}_{k+1|k+1} \tilde{x}_{k+1|k+1}^T]$$  \hfill (2.14)

With the defined framework it is possible to derive the filter’s dynamics. The objective of the filter is, in each iteration, to compute the state estimate $\tilde{x}_{k+1|k+1}$. In order to achieve our goal, let us assume that this vector is a linear combination of the prediction estimate $x_{k+1|k}$ and a weighted difference of the measurement $y_{k+1}$ and the measurement prediction $H_{k+1} \hat{x}_{k+1|k}$ (Simon, 2006):

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left[ y_{k+1} - H_{k+1} \tilde{x}_{k+1|k} \right]$$  \hfill (2.15)

The measurement innovation reflects the deviation of the estimated measurement from the actual measurement and is of most interest since it contains the new information about the states. The factor $K_{k+1}$ represents the Kalman Gain and is a weighting factor that aims to minimize the error covariance matrix $P_{k+1|k+1}$. The Kalman Gain $K_{k+1}$ that minimizes Eq. 2.14 can be written as (Grewal et al., 2001):

$$K_{k+1} = P_{k+1|k} H_{k+1}^T \left[ H_{k+1} P_{k+1|k} H_{k+1}^T + R_k \right]^{-1}$$  \hfill (2.16)

It is now possible to calculate the error covariance matrix update by using both Eq. 2.14 and the definitions in Eq. 2.8 and Eq. 2.9:

$$P_{k+1|k+1} = P_{k+1|k} - P_{k+1|k} H_{k+1}^T \left[ H_{k+1} P_{k+1|k} H_{k+1}^T + R_k \right]^{-1} H_{k+1} P_{k+1|k}$$  \hfill (2.17)

The filter must be initialized with given initial conditions for the initial state $x_0$ and covariance $P_0$.

**Prediction**

$$\hat{x}_{k+1|k} = A_k \hat{x}_k|k + B_k u_k$$  \hfill (2.18)

$$P_{k+1|k} = A_k P_{k|k} A_k^T + G_k Q_k G_k^T$$  \hfill (2.19)

**Filtering**

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left[ y_{k+1} - H_{k+1} \tilde{x}_{k+1|k} \right]$$  \hfill (2.20)

Eq. 2.16
\[ P_{k+1|k+1} = [I - K_{k+1} H_{k+1} P_{k+1|k}] \]  

**Initial Conditions**

\[
\begin{align*}
\dot{x} &= (1|0) = x_0 \\
P_{1|0} &= P_0
\end{align*}
\]  

- The Kalman Filter is a linear, discrete time, time-varying system which uses as system inputs \( u_k \) and as system measurements \( y_k \). The output of the filter is \( \hat{x}_{k+1|k+1} \).

- The error covariance matrix \( P_{k+1|k} \) is independent of \( Y_1^{k-1} \). This means that no set of measurements is of more interest than other to eliminate uncertainty about the state \( x_k \). The filter gain and the error covariance can be computed before the filter actually runs, which is not generally the case in the nonlinear filter \[ \text{[Ribeiro, 2004]} \]

### 2.1.3 Extended Kalman Filter

Although the definition of the Kalman filter lies on the linearity of the dynamics and measurements equations, it is possible to “extended” it to nonlinear systems. The Extended Kalman Filter (EKF) gives an approximation of the optimal estimate. The nonlinearity of the systems’s dynamics are approximated by a linearized version of the nonlinear system model around the last state estimate. Consider the nonlinear dynamics:

\[
\begin{align*}
x_k &= f_k(x_k, u_k, k) + G(x_k, k) w_k \\
y_k &= h_k(x_k, u_k) + v_k
\end{align*}
\]  

where both the system dynamics \( f_k(.) \) and the observer equation \( h_k(.) \) are nonlinear and assumed to be continuously differentiable. The assumptions made for the noise are the same as in Section 2.1.2.

With the nonlinear dynamics, the probability density functions seen before in Section 2.1.2 are non-Gaussian. To evaluate its first and second moments, the optimal nonlinear filter has to propagate the entire pdf which, in the general case, represents a heavy computational burden \[ \text{[Ribeiro, 2004]} \]. Therefore at each cycle, instead of propagating the entire non-Gaussian pdf, the EKF linearizes Eq. 2.23 and Eq. 2.24 around the last estimate of the state and uses a Kalman filter for the linearized dynamics.

The EKF may be resumed in the following steps:

Let \( F(k) \) and \( H(k) \) be the Jacobian matrices of \( f(.) \) and \( h(.) \):

\[
\begin{align*}
F_k &= \nabla f_k|x_{k|k} \\
H_k &= \nabla h_k|\hat{x}_{k+1|k}
\end{align*}
\]  

**Time Update (Prediction Cycle)**

1. One step ahead state prediction is obtain by integrating the nonlinear state of Eq. 2.23

\[
\hat{x}_{k+1|k} = \hat{x}_{k|k} + \int_k^{k+1} f(x_k, u_k, k) \, dk
\]
2. One step ahead error covariance matrix prediction:

\[ P_{k+1|k} = F_k P_{k|k} F_k^T + Q_k \]  \hspace{1cm} (2.28)

**Measurement Update (Correction Cycle)**

3. Kalman gain is as follows:

\[ K_{k+1} = P_{k+1|k} H_{k+1}^T \left( H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \right)^{-1} \]  \hspace{1cm} (2.29)

4. Update state estimate:

\[ \hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left[ y_{k+1} - h_{k+1}(\hat{x}_{k+1|k}) \right] \]  \hspace{1cm} (2.30)

5. Update error covariance matrix:

\[ P_{k+1|k+1} = \left[ I - K_{k+1} H_{k+1} \right] P_{k+1|k} \left[ I - K_{k+1} H_{k+1} \right]^T + K_{k+1} R_{k+1} K_{k+1}^T \]  \hspace{1cm} (2.31)

The error covariance matrix in the prediction cycle (Eq. 2.28) is calculated assuming the linearization of the system dynamics \( f_k(.) \) and is obtained similarly to the Kalman Filter. The Kalman Gain in the correction cycle (Eq. 2.29) assumes a linearization of the observer equation (seen in Eq. 2.26).

The EKF is not an optimal filter, but an implementation based on a set of approximations (first-order linearizations). Both linearizations \( F_{k+1} \) and \( H_{k+1} \) depend on previous state estimates and respective measurements. This implies that the covariance error matrix (for the update and correction step) and the filter gain have to be computed on-line in each iteration, which is not required for the Kalman filter.

If consecutive linearizations do not represent satisfying approximation by the linear model or if the estimated covariance matrix repeatedly underestimates the true covariance matrix, the EKF may diverge \cite{Simon,2006}. In order to avoid the underestimation of the true covariance matrix, usually a “stabilizing noise” is added \cite{Ribeiro,2004}.

**2.1.4 Robust Extended Kalman Filter**

One of the main assumptions of both the EKF and KF is that a dynamic model of the system considered is exactly known (or at least - in the EKF case - is assumed that the first-order term is an approximation good enough so that the filter does not diverge). Since the filter type may not be robust against this uncertainty, over the past few years more methods were developed to satisfy this need. With the same goal, in \cite{Haddad,1987}, a discrete-time state estimator was developed with guaranteed cost bounds for linear systems using parametric uncertainties (the parameter uncertainty was modeled by both the state and the measurement noise).

In the Robust Extended Kalman Filter (REKF) a Kalman filter design is considered for linear discrete-time systems with a norm-bounded parameter uncertainty in both the state and output matrices and can be adapted for non-linear systems. The algorithm presented here is based on \cite{Einicke,1999} and \cite{Xie,1994} and considers the general nonlinear case defined before in Eq. 2.23 and Eq. 2.24. The approach taken in the Robust EKF is not to neglect the higher order terms of the Taylor series expansions but rather assume them to be functions of the state estimation error and the exogenous

\[ \text{[11]} \]
inputs which have bounded $H_\infty$. This approach leads to a minimax estimation problem that can be treated using standard $H_\infty$ methods (Einicke, 1999).

Once again is assumed that functions $f(\cdot)$ and $h(\cdot)$ are continuously differentiable and the noise processes $w_k$ and $v_k$ are sequences of zero mean, white Gaussian noise with covariances matrices $Q_k$ and $R_k$ respectively.

**Time Update (Prediction Cycle)**

1. One step ahead state prediction:
   \[
   \hat{x}_{k|k-1} = f_k(\hat{x}_{k-1|k-1})
   \]  
   (2.32)

2. One step ahead error covariance matrix prediction:
   \[
   P_{k|k-1} = F_k P_{k-1} F_k^T + Q_k
   \]  
   (2.33)

3. One step ahead covariance matrix prediction:
   \[
   \Sigma_{k|k-1} = \left( P^{-1}_{k-1} - \gamma^{-2} L_k^T L_k \right)^{-1}
   \]  
   (2.34)

**Measurement Update (Correction Cycle)**

4. An auxiliary matrix is defined:
   \[
   P_{y,k} = H_k \Sigma_{k|k-1} H_k^T + R_k
   \]  
   (2.35)

5. The Kalman Gain estimate is:
   \[
   K_k = \Sigma_{k|k-1} H_k^T P_{y,k}^{-1}
   \]  
   (2.36)

6. The estimate of state is as follows:
   \[
   \hat{x}_k = \hat{x}_{k|k-1} + K_k \left[ y_k - h(\hat{x}_{k|k-1}) \right]
   \]  
   (2.37)

7. The final estimation of the error covariance matrix is given by:
   \[
   P_k = \left( \Sigma_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k \right)^{-1}
   \]  
   (2.38)

where both $F_k = \frac{\partial f(x)}{\partial x} \bigg|_{x=\hat{x}_{k-1|k-1}}$ and $H_k = \frac{\partial h(x)}{\partial x} \bigg|_{x=\hat{x}_{k|k-1}}$ are the Jacobian matrix of $f(x_{k-1})$ and $h(x_k)$ respectively. $L_k$ represents the identity matrix $I$ (later on in the AREKF this parameter will be redesigned). The tuning parameter $\gamma$ is non-zero and is set in order to maintain $\Sigma_{k|k-1}$ as a positive definite matrix. By definition, the REKF objective is to guarantee that the attenuation level $\gamma$ is always greater than the norm of the transfer function between the estimation error and the external disturbances (modeling errors and system noises)

\[
\gamma^2 \geq \frac{\| L_k \tilde{x}_k \|^2}{\| w_k \|^2 + \| \Delta_k \|^2 + \| v_k \|^2}
\]  
(2.39)

where $\tilde{x}$ is the estimation error ($\tilde{x}_k = x_k - \hat{x}_k$), $w_k$ and $v_k$ are the noise vectors (Eq. 2.23 and Eq. 2.24) and $\Delta_k$ is the model error caused by unknown exogenous inputs or by the linearization error.

Given the structure of the REKF it is easily shown that the REKF tends to a standard EKF when $\gamma \to \infty$ (see Eq. 2.34). This means that $\gamma$ may be interpreted as a tuning parameter to control the trade-off between a minimum variance performance and $H_\infty$ performance.
2.1.4.A Stability Analysis

In order to further understand the REKF algorithm and to be able to modify it to an “Adaptive” version, a closer look to the stability of the filter is essential. The stability analysis rests on the following lemma [Einicke, 1999]

Lemma 2.1: Assuming there is a stochastic process $V(x_k)$ (being $x_k$ a stochastic process exponentially bounded in mean square\(^3\)) as well as real numbers $v_{min}, v_{max}, \mu > 0$ and $0 < \lambda < 1$ such that:

$$v_{min} \| \xi_k \|^2 \leq V(x_k) \leq v_{max} \| \xi_k \|^2$$  \hspace{1cm} (2.40)

and

$$E\{V(x_k) \xi_{k-1} \} - V(x_{k-1}) \leq \mu - \lambda V(x_{k-1})$$ \hspace{1cm} (2.41)

are fulfilled. The energy of $\xi_k$ is represented by a Lyapunov function $V(x_k)$ and is user defined. A proper choice of $V(x_k)$ may - under some conditions - ease the analysis [Freeman, 2008] The upper and lower bound of $V(x_k)$ are defined by $v_{max}$ and $v_{min}$ respectively.

Then the stochastic process $x_k$ is exponentially bounded in mean square (proof of Lemma 2.1 can be seen in Appendix A.3):

$$E\{ \| \xi_k \|^2 \} \leq \frac{v_{max}}{v_{min}} E\{ \| \xi_{0} \|^2 \} (1 - \lambda)^k + \frac{\mu}{v_{min}} \sum_{i=1}^{k-1} (1 - \lambda)^i$$ \hspace{1cm} (2.42)

The condition in Eq. 2.41 forces the energy of $\xi_k$ not increasing arbitrarily when it bounds the process $x_k$.

Defining the prediction error of the REKF similarly to the KF ($\tilde{x}_k|_{k-1} = x_k - \hat{x}_k|_{k-1}$) and substituting Eq.2.23 and Eq. 2.32 (assuming $G_k = I$ for simplicity) into the prediction error, one gets:

$$\tilde{x}_{k|k-1} = \beta_k F_k \tilde{x}_{k-1} + w_k$$ \hspace{1cm} (2.43)

where $\beta$ is a diagonal, unknown and time-varying matrix. The real error covariance matrix of the prediction error can be approximated by:

$$\Sigma_{k|k-1} = E(\tilde{x}_{k|k-1} \tilde{x}_{k|k-1}^T) = E\left[(\beta_k F_k \tilde{x}_{k-1} + w_k)(\beta_k F_k \tilde{x}_{k-1} + w_k)^T\right]$$

$$\beta_k F_k P_{k-1} F_k^T \beta_k + Q_k$$ \hspace{1cm} (2.44)

The $\beta$ matrix is used to further estimate the prediction error caused by the linearization error and the unknown exogenous inputs. The following theorem uses these prerequisites as sufficient conditions and ensures the stability of the REKF [Reif, 1994]

Theorem 2.1: The theorem states that if the three following conditions are fulfilled:

1. There are numbers $\{f_{min}, f_{max}, \beta_{min}, \beta_{max}, h_{min}, h_{max}\} \in \mathbb{R}$ such that:

$$f_{min}^2 I \leq F_k F_k^T \leq f_{max}^2 I$$

$$\beta_{min}^2 I \leq \beta_k \beta_k^T \leq \beta_{max}^2 I$$

$$h_{min}^2 I \leq H_k H_k^T \leq h_{max}^2 I$$ \hspace{1cm} (2.45)

2. There are numbers $\{p_{min}, p_{max}, r_{min}, r_{max}, q_{max}\} \in \mathbb{R}$

$$p_{min}^2 I \leq P_k \leq p_{max}^2 I$$

$$r_{min}^2 I \leq R_k \leq r_{max}^2 I$$

$$Q_k \leq q_{max} I$$ \hspace{1cm} (2.46)

\(^3\)The stochastic process $x_k$ is said to be exponentially bounded in mean square if there are real numbers $\eta, \nu > 0$ and $0 < \phi < 1$ such that $E|\| \xi_n \|^2 \| \leq \eta \| \xi_0 \|^2 \phi^n + \nu$ holds true for every $n \geq 0$.  

13
3. The following matrix in-equality is fulfilled

\[ \Sigma_{k|k-1} > \bar{\Sigma}_{k|k-1} \]  

(2.47)

This condition can also be interpreted as \( \Sigma_{\delta|\delta-1} > \bar{\Sigma}_{\delta|\delta-1} \) being a positive-definite matrix. Then the following condition holds true:

\[
E\{\|\tilde{x}_k\|^2\} \leq \frac{p_{\max}}{p_{\min}} E\{\|\tilde{x}_0\|^2\}(1 - \lambda_{\min})^k + \frac{p_{\max}}{p_{\min}} \sum_{i=1}^{k-1} (1 - \lambda_{\min})^i
\]  

(2.48)

Assuming \( p_{\max} > 0 \) and \( 0 < \lambda_{\min} \leq 1 \). The proof of this Theorem is in Appendix A. Throughout the proof the measurement equation is assumed to be linear to simplify the deduction, but the analysis remains valid if the linearization error of the measurement equation is negligible - analogous to the EKF - (Xiong, 2009). This Theorem yields several properties:

- The estimation error remains bounded and the effect of the initial estimation error reduces with iterations.
- The first condition of Eq. 2.46 is further discussed in (Reif, 1994) and relates to the observability property of the linearized system.
- The third condition in Eq. 2.47 is a crucial condition. It states that the calculated covariance matrix should be larger than the real one, which traditionally enhances the filter stability (Emicke, 1999 and Simon, 2006). This property leads to a potential problem: matrix \( \beta_k \) in the presence of unknown inputs may be rather large. This propagates to the prediction error (Eq. 2.44) which may violate the condition in Eq. 2.47. The best solution to this problem is to tune the parameter \( \gamma \) that, in order to \( \Sigma_{k|k-1} - \bar{\Sigma}_{k|k-1} \) be positive, must be:

\[
(P_{k|k-1}^{-1} - \gamma^{-2} I)^{-1} - \bar{\Sigma}_{k|k-1} > 0
\]

\[
\gamma^{-2} > \max\{\text{eig}(P_{k|k-1}^{-1} - \bar{\Sigma}_{k|k-1}^{-1})\}
\]  

(2.49)

Assuming \( L_k = I \). If \( \max\{\text{eig}(P_{k|k-1}^{-1} - \bar{\Sigma}_{k|k-1}^{-1})\} \leq 0 \), condition in Eq. 2.49 is certainly fulfilled. Otherwise, \( \gamma \) should satisfy:

\[
\gamma < \max\{\text{eig}(P_{k|k-1}^{-1} - \bar{\Sigma}_{k|k-1}^{-1})\}^{-\frac{1}{2}}
\]  

(2.50)

Which means \( \gamma \) should be as small as possible, to improve the stability of the algorithm. This tendency is consistent with the robust \( H_{\infty} \) filter that states the attenuation level \( \gamma \) should be as small as possible (Souza, 2002 and Simon, 2006).

- At the same time, for the error covariance matrix \( \Sigma_{\delta|\delta-1} \) to be positive definite:

\[
P_{k|k-1}^{-1} - \gamma^{-2} I > 0 \Rightarrow \gamma > \{\max\{\text{eig}(P_{k|k-1})\}\}^{\frac{1}{2}}
\]  

(2.51)

If both conditions for \( \gamma \) in Eq. 2.50 and Eq. 2.51 can not be simultaneously satisfied (which means \( \{\max\{\text{eig}(P_{k|k-1})\}\}^{1/2} > \{\max\{\text{eig}(P_{k|k-1}^{-1} - \bar{\Sigma}_{k|k-1}^{-1})\}\}^{1/2} \)) the stability of the REKF is not guaranteed.

This tuning of the attenuation level \( \gamma \) implies that the ability of the REKF to minimize the energy of the estimation error is limited by the maximum eigenvalue of \( P_{k|k-1} \). From Eq. 2.39 for a fixed
value of $\gamma$ the bound of the estimation error $\| L_k \tilde{x}_k \|^2$ is enlarged by the presence of a linearization error or unknown exogenous inputs. Moreover, a large deviation of the estimated state from the real one will increase the linearization error. If this tendency does not stop, the filter will not converge.

2.1.5 Adaptive and Robust Extended Kalman Filter

Since the REKF may not converge at all times, a new method is proposed to design an Adaptive Robust Extended Kalman Filter (AREKF). The REKF algorithm (in Theorem 2.1) states that the bound of $\tilde{x}_k$ is manageable by enlarging the calculated covariance matrix $\Sigma_{k|k-1}$, which enlarges by decreasing $\gamma$. This represents a problem, since under some conditions it may not be possible to tune $\gamma$ such that condition in Eq. 2.47 holds true for $L_k = I$. Redesigning matrix $L_k$ solves this problem (Xiong, 2009):

$$ L_k = \gamma \left( P_{k|k-1}^{-1} - \lambda_k^{-1} P_{k|k-1}^{-1} \right)^{1/2} \quad (2.52) $$

where $\lambda_k$ is a tuning parameter that should be large enough to satisfy the following condition:

$$ \bar{\Sigma}_{k|k-1} < \lambda_k P_{k|k-1} \quad (2.53) $$

Using both Eq. 2.52 and Eq. 2.34 it is possible to conclude that condition 2.47 of Theorem 2.1 holds true. This way one avoids to tune the $\gamma$ parameter but instead must tune $\lambda_k$, to obtain a better robust behaviour (Xiong, 2009). In the next chapter further details about the $\lambda_k$ tuning are given.

In spite of the increase of degrees of freedom, the upper bound of $\lambda_k P_{k|k-1}$ may be too conservative, since there is a loss of optimality of the algorithm trying to accommodate the worst case (largest linearization error). In order to not decrease the accuracy and to improve the stability at the same time, an adaptive scheme to adjust $\Sigma_{k|k-1}$ is proposed:

$$ \Sigma_{k|k-1} = \begin{cases} P_{k|k-1}^{-1}, & \text{tr}(P_{y,k}) > \alpha \text{tr}(\bar{P}_{y,k}) \\ \left( P_{k|k-1}^{-1} - \gamma^{-2} L_k^T L_k \right)^{-1}, & \text{otherwise} \end{cases} \quad (2.54) $$

where $\bar{P}_{y,k} = E(\tilde{y}_k \tilde{y}_k^T | \tilde{x}_{k-1})$ is the real covariance matrix of the innovation $\tilde{y}_k = y_k - H_k \hat{x}_{k|k-1}$. The parameter $\alpha (> 0)$ has to be tuned during the implementation process and represents a threshold between both behaviours of $\Sigma_{k|k-1}$. Despite being unknown, $\bar{P}_{y,k}$ can be estimated by (Xiong, 2009):

$$ \bar{P}_{y,k} \approx \begin{cases} \tilde{y}_k \tilde{y}_k^T, & k = 0 \\ \rho \bar{P}_{y,k-1} + \tilde{y}_k \tilde{y}_k^T, & k > 0 \end{cases} \quad (2.55) $$

where $\rho$ is a forgetting factor, also defined in the implementation section.

The structural difference between the REKF and the AREKF is that the prediction error covariance follows Eq. 2.54. The implication of this modification is that, when the innovation is considerable, the prediction error covariance matrix will be set to $\left( P_{k|k-1}^{-1} - \gamma^{-2} L_k^T L_k \right)^{-1}$ to avoid filter divergence and, when the innovation is small, it will be set to the previous one, so that the estimation is not distorted.

2.1.5.A Stability Analysis

Similarly to the REKF, the stability of the AREKF is fully characterized on the following Theorem (Xiong, 2008):
Theorem 2.2: Assuming a nonlinear stochastic system with linear measurement, \( \text{rank}(H_k) = l \), \( \alpha = 1 \) and real error covariance matrix approximated by \( \bar{\Sigma}_{k|k-1} \), the Eqs. 2.32, 2.33, 2.37 - 2.36, 2.52 and 2.54 define the AREKF. If the two following conditions are fulfilled:

1. There are numbers \( \{f_{\text{min}}, f_{\text{max}}, \beta_{\text{min}}, \beta_{\text{max}}, h_{\text{min}}, h_{\text{max}}\} \in \mathbb{R} \) such that:
   \[
   f_{\text{min}} I \leq F_k F_k^T \leq f_{\text{max}} I, \quad \beta_{\text{min}} I \leq \beta_k \beta_k^T \leq \beta_{\text{max}} I, \quad h_{\text{min}} I \leq H_k H_k^T \leq h_{\text{max}} I
   \] (2.56)

2. There are numbers \( \{p_{\text{min}}, p_{\text{max}}, r_{\text{min}}, r_{\text{max}}, q_{\text{max}}\} \in \mathbb{R} \)
   \[
   p_{\text{min}} I \leq P_k \leq p_{\text{max}} I, \quad r_{\text{min}} I \leq R_k \leq r_{\text{max}} I, \quad Q_k \leq q_{\text{max}} I
   \] (2.57)

Then the following condition holds true for real numbers \( \mu_{\text{max}} > 0 \) and \( 0 \leq \lambda_{\text{min}} \leq 1 \):

\[
E\{\|\tilde{x}_k\|^2\} \leq \frac{p_{\text{max}}}{p_{\text{min}}} E\{\|\tilde{x}_0\|^2\}(1 - \lambda_{\text{min}})^k + \frac{\mu_{\text{max}}}{p_{\text{min}}} \sum_{i=1}^{k-1} (1 - \lambda_{\text{min}})^i
\] (2.58)

The proof of Theorem 2.2 is in the Appendix A. The properties of the AREKF can be summarized as follows:

- The main idea of the AREKF is to enhance the REKF, designing a filter based on the stability analysis. This translates in determining whether the error covariance matrix \( (\Sigma_{k|k-1}) \) should be reset. The main practical difference between both algorithms, is that the AREKF proposes a design based on the tuning of the \( \lambda_k \) parameter instead of the attenuation level \( \gamma \). This represents an enhancement of efficiency and a prevention of instabilities in case of large prediction errors since it may be impossible to obtain an appropriate \( \gamma \) for the REKF.

- The AREKF alternates between the standard EKF controlled by the innovation covariance \( \bar{P}_{y,k} \) and the REKF when the estimated covariance \( \bar{P}_{y,k} \) exceeds the threshold set by the \( \alpha \) parameter. More accuracy is expected from this algorithm than the traditional REKF [Xiong, 2009].

- The assumption of \( \text{rank}(H_k) = l \) is a necessary condition for the theorem, however it is possible to use the AREKF even if \( \text{rank}(H_k) < l \) (this implies that the condition in Eq. 2.47 is violated). This means that it is advisable to adopt the robust filtering to adjust the covariance matrix \( \Sigma_{k|k-1} \) when \( \text{rank}(H_k) < l \) and the stability condition is not satisfied.

- In both REKF and AREKF condition \( \Sigma_{k|k-1} > \bar{\Sigma}_{k|k-1} \) is sufficient but not necessary to ensure filter stability. Note that parameter \( \alpha \) is introduced so that a reset of the covariance matrix \( \Sigma_{k|k-1} \) is avoided. Numerical simulations have shown [Xiong, 2009] that even if the condition \( \Sigma_{k|k-1} > \bar{\Sigma}_{k|k-1} \) is not globally satisfied as \( \alpha < 1 \), the AREKF is still more robust to higher prediction error than the REKF or the traditional EKF alone.

- The estimate of \( \bar{P}_{y,k} \) is never used directly in the algorithm, but as an indicator of the filter stability. Even if it is badly estimated using Eq. 2.55 and the covariance matrix is reset inappropriately it will not affect the stability of the filter.

- It is possible to extend this analysis to a system with disturbances in the measurement equations, and the innovation covariance matrix will follow the approach taken here of the error covariance
matrix, to ensure filter stability. New problems arise for the stability analysis when both the state and the measurement equation are nonlinear, and the design for both matrices (innovation covariance and error covariance) would require further investigation.

- The conclusions and derivation of Theorem 2.1 and 2.2 can be extended for the Unscented Kalman Filter (UKF) with the proof suffering minor changes to the one shown in the Appendix for the REKF and AREKF.
2.2 Reference Frames

2.2.1 Earth-Centered Inertial Coordinate System

The Earth-Centered Inertial (ECI) coordinate system has its origin at the center of mass of the Earth with the axes pointing in fixed directions with respect to the stars. In ECI coordinate system, the xy-plane is taken to coincide with the Earth’s equatorial plane, the +x-axis is permanently fixed in a particular direction relative to the celestial sphere, the +z-axis is taken normal to the xy-plane in the direction of the north pole, and the +y-axis is chosen so as to form a right-handed coordinate system.

Since the x-axis is defined relative to the celestial sphere and the z-axis is defined relative to the equatorial plane a problem arises, since the equatorial plane moves with respect to the celestial sphere: largely due to the Earth’s shape and the gravitational pull of the Sun and the Moon, which would make the ECI not to be truly inertial \cite{Kaplan, 2005}. To solve this, the orientation of the axes is defined at a precise instant in time (epoch). The most used in GNSS applications uses the orientation of the equatorial plane at 12:00 (UTC) on January 1, 2000, called J2000 system. In this reference frame, the x-axis points the direction of the vernal equinox (from the center of mass of the Earth) and both the y- and z- axes are defined as before. With the orientation of the axes being fixed, the ECI can be considered inertial.

Since a GNSS satellite obeys Newton’s laws of motion and gravitation in an ECI coordinate system, for measuring and determining the orbits of the GNSS satellites, it is beneficial to use an ECI coordinate system.

2.2.2 Earth-Centered Earth-Fixed Coordinate System

If the intent is computing the position of a GNSS receiver, it is more convenient to use a coordinate system that rotates with the Earth, known as an Earth-centered Earth-fixed (ECEF) system.

The ECEF coordinate system used for GNSS has its xy-plane coincident with the Earth’s equatorial plane, as the ECI frame. However, in the ECEF system, the +x-axis points in the direction of 0° longitude, and the +y-axis points in the direction of 90° East longitude. The x-, y-, and z-axes rotate with the Earth and no longer describe fixed directions in inertial space. In this ECEF system, the z-axis is chosen to be normal to the equatorial plane in the direction of the geographical North Pole, completing the right-handed coordinate system.

It is possible to transform the coordinates between ECI and the ECEF systems. These transformations are accomplished by the application of rotation matrices to the satellite position and velocity vectors in the ECI coordinate system. With the satellite position and velocity in the ECEF frame, it is possible to formulate the entire navigation problem in the ECEF system avoiding the details of the orbit determination or the transformation to the ECEF system \cite{Parkinson, 1996}.

\footnote{Exception made for the Sagnac effect on signal propagation in the rotating frames.}
2.3 Fundamentals of Satellite Navigation

A Global Positioning System (GPS) receiver needs accurate information about the positions of the GPS satellites in order to determine its position. Hence, understanding how the GPS orbits are characterized becomes of most importance.

2.3.1 Orbital Mechanics

The most significant force acting on a satellite is the Earth’s gravitation. According to Newton’s laws, assuming a perfectly spherical and of uniform density Earth can be written as:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r}$$  \hspace{1cm} (2.59)

where $\mu$ is the product of the mass of the Earth and the Gravitational constant and, in datum WGS-84, values $3986005 \times 10^8 \text{m}^3/\text{s}^2$. Eq. (2.59) is the so-called two-body equation or Keplerian satellite motion. Since the Earth is not spherical and the mass distribution is not homogeneous, this does not represent the true acceleration of the satellite induced by the Earth’s gravitation. A more general model may be derived considering a general gravitational potential ($V$) and solving the general case of

$$\frac{d^2\mathbf{r}}{dt^2} = \nabla V$$  \hspace{1cm} (2.60)

that yields [Kaplan, 2005] a gravitational potential of

$$V = \frac{\mu}{r} \left[ 1 + \sum_{l=2}^{\infty} \sum_{m=0}^{l} \left( \frac{a}{r} \right)^l P_{lm}(\sin\phi')(C_{lm}\cos m\alpha + S_{lm}\sin m\alpha) \right]$$  \hspace{1cm} (2.61)

where $r$ is the distance of P from the origin, $\phi'$ is the geocentric latitude of P, $\alpha$ is the right ascension of P, $a$ is the mean equatorial radius of the Earth, $P_{lm}$ is the associated Legendre function, $C_{lm}$ is the spherical harmonic cosine coefficient (degree $l$, order $m$) and $S_{lm}$ is the spherical harmonic sine coefficient (degree $l$, order $m$).

In order to find the position and velocity vectors of a satellite on a two-body orbit (Eq. 2.59) in an arbitrary point in time, one only needs the six integrals of motion and the initial time. For the perturbed equation of motion (Eq. 2.61), it is also necessary to characterize how these parameters change in time. This is the approach of the GPS ephemeris message and allows the receiver to calculate its corrected position and velocity.

The GPS uses a particular set (there are many possible formulations) of six integrals of motion known as the Keplerian orbital elements. Three elements define the shape of the satellite’s orbit and the other three its orientation. Fig. 2.1 illustrates the three orbital elements that define the shape of the orbit. The focus point is at point $F$, which represents the center of mass of the Earth and the...
elliptical orbit has semimajor axis $a$ and eccentricity $e$. Along with the time at which the satellite passes perigee $\tau$, these are the three elements that define the shape of the orbit.

Figure 2.1: The three Keplerian orbital elements defining the shape of the satellite's orbit (Kaplan, 2005)

In practice the GPS system does not directly use the time of perigee passage but an equivalent, that is known as the mean anomaly at epoch[^1]. The mean anomaly is the angle $\nu$ in Fig. 2.1 and is the angle that relates to the true anomaly at epoch.

The other three orbital elements, that define the orientation of the orbit are presented in Fig. 2.2. The GPS system uses Keplerian elements that are defined in the ECEF coordinate system (Sec. 2.2). The three elements are: inclination of orbit ($i$), longitude of the ascending node ($\Omega$) and the argument

[^1]: Epoch is an instant in time, in this case means the time at point $A$
of perigee ($\omega$).

The inclination is the angle between the Earth’s equatorial plane and the satellite’s orbital plane.

Being the ascending node the point in the satellite’s orbit where it crosses the equatorial plane, the other two orbital elements are defined in relation to this point. The angle between the direction of the ascending node and the x-axis is called right ascension of the ascending node and in the ECEF system case (the x-axis is aligned with the meridian of 0° longitude) this angle is the longitude of the ascending node. This angle is measured in the equatorial plane.

Finally the argument of perigee expresses the angle between the ascending node and the orbit’s perigee. This angle is measured in the orbital plane.

The typical GPS satellite’s orbit is almost circular, with eccentricities of about 0.02 (maximum), semimajor axes of 26560 km and orbital period of 43,080 seconds (almost 12 hours). The inclination is about 55° and the remaining orbital parameters vary in a way that the satellites are geometrically distributed around the Earth, in order to maximize the coverage.

Table 2.2 summarizes the elements and variables used for the ephemeris computation, in order to find the satellite’s position ([Kaplan, 2005]).

Table 2.3 provides the algorithm used by the GPS receiver to compute the position vector of a satellite in the ECEF coordinate system from the orbital elements in Table 2.2.

There are some aspects that should be detailed from Table 2.3. On step (5) we come across the Kepler’s equation. This is a solvable transcendental equation in the $E_k$ parameter and the solution should be found numerically (usually even the Newton’s method is enough for a satisfactory computational error). In (6) a special attention to the quadrant of the result must be taken. Computing (14) the rotation rate of the Earth should be set to $\Omega_e = 7.2921151467 \times 10^{-5}$ rad/s, according to IS-GPS-2000 (also consistent with WGS84).
Table 2.2: GPS Ephemeris data definitions.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{oe}$</td>
<td>Reference time of ephemeris</td>
</tr>
<tr>
<td>$\sqrt{a}$</td>
<td>Square root of semimajor axis</td>
</tr>
<tr>
<td>$e$</td>
<td>Eccentricity</td>
</tr>
<tr>
<td>$i_0$</td>
<td>Inclination angle (at time $t_{oe}$)</td>
</tr>
<tr>
<td>$\Omega_0$</td>
<td>Longitude of the ascending node (at weekly epoch)</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Argument of perigee (at time $t_{oe}$)</td>
</tr>
<tr>
<td>$M_0$</td>
<td>Mean anomaly (at time $t_{oe}$)</td>
</tr>
<tr>
<td>$di/dt$</td>
<td>Rate of change of inclination angle</td>
</tr>
<tr>
<td>$\dot{\Omega}$</td>
<td>Rate of change of the longitude of the ascending node</td>
</tr>
<tr>
<td>$\Delta n$</td>
<td>Mean motion correction</td>
</tr>
<tr>
<td>$C_{uc}$</td>
<td>Amplitude of cosine correction to argument of latitude</td>
</tr>
<tr>
<td>$C_{us}$</td>
<td>Amplitude of sine correction to argument of latitude</td>
</tr>
<tr>
<td>$C_{rc}$</td>
<td>Amplitude of cosine correction to orbital radius</td>
</tr>
<tr>
<td>$C_{rs}$</td>
<td>Amplitude of sine correction to orbital radius</td>
</tr>
<tr>
<td>$C_{ic}$</td>
<td>Amplitude of cosine correction to inclination angle</td>
</tr>
<tr>
<td>$C_{is}$</td>
<td>Amplitude of sine correction to inclination angle</td>
</tr>
</tbody>
</table>

Table 2.3: Computation of a Satellite’s ECEF Position Vector (Kaplan, 2005)

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a = (\sqrt{a})^2$</td>
<td>Semimajor axis</td>
</tr>
<tr>
<td>2</td>
<td>$n = \sqrt{\frac{\mu}{a^3}} + \Delta n$</td>
<td>Corrected mean motion</td>
</tr>
<tr>
<td>3</td>
<td>$t_k = t - t_{oe}$</td>
<td>Time from ephemeris epoch</td>
</tr>
<tr>
<td>4</td>
<td>$M_k = M_0 + n(t_k)$</td>
<td>Mean anomaly</td>
</tr>
<tr>
<td>5</td>
<td>$M_k = E_k - e \sin E_k$</td>
<td>Eccentric anomaly (is solved iteratively for E)</td>
</tr>
<tr>
<td>6</td>
<td>$\sin u_k = \frac{\sqrt{1 - e^2} \sin E_k}{1 - e \cos E_k}$</td>
<td>True anomaly</td>
</tr>
<tr>
<td>7</td>
<td>$\phi_k = u_k + \omega$</td>
<td>Argument of latitude</td>
</tr>
<tr>
<td>8</td>
<td>$\delta \phi_k = C_{us} \sin(2\phi_k) + C_{uc} \cos(2\phi_k)$</td>
<td>Argument of latitude correction</td>
</tr>
<tr>
<td>9</td>
<td>$\delta r_k = C_{rs} \sin(2\phi_k) + C_{rc} \cos(2\phi_k)$</td>
<td>Radius correction</td>
</tr>
<tr>
<td>10</td>
<td>$\delta i_k = C_{is} \sin(2\phi_k) + C_{ic} \cos(2\phi_k)$</td>
<td>Inclination correction</td>
</tr>
<tr>
<td>11</td>
<td>$u_k = \phi_k + \delta \phi_k$</td>
<td>Corrected argument of latitude</td>
</tr>
<tr>
<td>12</td>
<td>$r_k = a(1 - e \cos E_k) + \delta r_k$</td>
<td>Corrected radius</td>
</tr>
<tr>
<td>13</td>
<td>$i_k = i_0 + (di/dt) t_k + \delta i_k$</td>
<td>Corrected inclination</td>
</tr>
<tr>
<td>14</td>
<td>$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e) t_k - \dot{\Omega}<em>e t</em>{oe}$</td>
<td>Corrected longitude of node</td>
</tr>
<tr>
<td>15</td>
<td>$x_s = r_k \cos u_k \cos \Omega_k - r_k \sin u_k \sin i_k \sin \Omega_k$</td>
<td>ECEF x-coordinate</td>
</tr>
<tr>
<td>16</td>
<td>$y_s = r_k \cos u_k \sin \Omega_k + r_k \sin u_k \sin i_k \cos \Omega_k$</td>
<td>ECEF y-coordinate</td>
</tr>
<tr>
<td>17</td>
<td>$z_s = r_k \sin u_k \sin i_k$</td>
<td>ECEF z-coordinate</td>
</tr>
</tbody>
</table>
2.3.1.A Orbital Design

There is a fairly big amount of combinations of orbital parameters for a satellite and it is convenient to arrange orbits into categories. This categorization can be made according to different orbital parameters. One possible categorization is by altitude:

- Low Earth Orbit (LEO) is a class of orbits with altitude typically less than 1,500 km. The orbital velocity needed to a stable orbit (height of 1,500 km) is $7.12 \text{ km/s}$. The International Space Station, for instance, is in a LEO orbit of around 400 km. The main advantages of this kind of orbit is the low energy necessary to place a satellite in this orbit, as well as the lower energy for signal transmission;

- Geostationary Earth Orbit (GEO) is an orbit with a period equal to the duration of the sidereal day. This means (Kepler equations) a semimajor axis of $a = 42,164.17 \text{ km}$ and an altitude of $35,786 \text{ km}$. Communications and weather satellites often use this orbit;

- Medium Earth orbit (MEO) is a class of orbits with altitudes below GEO and above LEO, with most practical examples being in the range of roughly $10,000 - 25,000 \text{ km}$ altitude.

Another possible categorization of orbits is by eccentricity:

- Circular orbits, that have (near) zero eccentricity;

- Highly Elliptical Orbit (HEO) that have large eccentricities (usually $e > 0.6$). The Molniya satellite is an example of satellite that uses this kind of orbit;

The categorization of orbits by inclination is also very common, but of no interest in this thesis. Note that there is a common misconception between two different terms that should be avoided. GEO defines an orbital altitude that allows that the period of the orbit equals the period of rotation of the Earth - the sidereal day. A geostationary orbit is a GEO orbit with zero inclination and zero eccentricity. To an observer on Earth - often called ground tracks - a satellite in geostationary orbit has no apparent motion because the relative position vector from the observer to the satellite remains fixed over time.

2.3.1.B Simplified Perturbations Models

Space propagation models use state information of a satellite in a certain time to predict a future state of the satellite. In a simplistic way, these models allow to use information of the position and the velocity now and make a reasonable guess where the satellite will be in the future. The satellite however, encounters disturbances, or perturbations, along its path that complicates its motion prediction. These perturbations are caused by the Earth’s shape (as in, spherical harmonics), effects from other bodies - the Sun and Moon are the more relevant - atmospheric drag, radiation, and others.

Propagation models are primarily used by agencies that track orbiting objects. In a perfect scenario, it would be possible to know the location of all the space objects at all times, using only a measurement
of the position and velocity of an object, and then resorting to the propagator to determine its location in the future.

The first model was made by [Kozai, 1959] by the National Space Surveillance Control Center and was soon ‘replaced’ by the Simplified Perturbations Models [SGP] model, in 1960. In the late 60’s SPACETRACK published SGP4 in the Report 3, however the official code, maintained by the United States government, became an export controlled black-box. Improvements to the code for better computing efficiency, more accurate constants, better programming techniques were never made public. Alternatively, these codes were made available for selected agencies.

Non-selected agencies that needed customized code for projects (NASA included) were forced to improve the original code found in the Report 3, leading to a wide variety of propagation codes based on the same model, with different improvements. Vallado, in 2006, reconciled the many codes into a single standardized code. This code was made available to the public through Celestrak.

The propagation routine for the SGP4 orbital model relies on the NORAD Two-Line Elements (TLE) data. This algorithm introduces an error of about 1 – 3 km per day [Vallado, 2006].
3

Modeling and Implementation

Contents

3.1 Orbit Generation .................................................. 26
3.2 Filter Implementation ........................................... 29
This chapter intends to cover the main features of the implementation techniques used in this work. First, the generation of the simulated orbits, as well as the GPS constellation are presented, in Section 3.1. Later in Section 3.2 the Equations of Motion will be derived in detail. The receiver clock is also analyzed, as well as the dynamic and observation models. This same section finishes with a summary of the modeled noise generation problem and a brief characterization of the tuned parameters.

3.1 Orbit Generation

3.1.1 GPS Constellation Generation

The GPS constellation was generated using a System Effectiveness Model (SEM) Almanac, from [Celestrak, 2000] using the algorithm in the Table 2.3. The Almanac used is dated March 2015 (SEM Almanac 0810.233472).

The entire 32 satellite’s constellation through an entire period is shown in Fig. 3.1. The position is presented in ECEF coordinates, each satellite with an associated colour.

![GPS Constellation Satellites in ECEF coordinates.](image)

3.1.2 Reference Orbits Generation

The simulated orbits for the GPS receiver were calculated using a TLE set. The conversion from TLE to ECI coordinates was made with the Vallado’s method and using the simplified perturbations model SGP4 seen in Subsection 2.3.1.B. Afterwards, a conversion from ECI to ECEF coordinates was made. This conversion uses the information of the position of the Earth in a certain date, therefore, the date of the coordinates is of most importance in this conversion; in this case, the Julian Date was
The generated orbit is used as reference in the three selected algorithms. Fig. 3.2 present the final trajectory for the LEO reference orbits, with a period $T = 5923\,s$ and a mean altitude of $\sim 700\,km$.

![Figure 3.2: Reference LEO orbit simulated in ECEF coordinates.](image)

Besides this orbit, other scenarios were also considered, to enable a broad number of testing environments.

First, for a LEO orbit, after the filter convergence, outliers in the receiver measurements were introduced. These outliers simulate real instrument errors that occasionally occur in GNSS receivers (e.g., pseudoranges and Doppler). This was simulated by multiplying the measurement pseudorange vector by factor of $(1 + 2 \times 10^{-6})$ at $t = 150\,s$ and $t = 151\,s$. This corresponds to an error of around 50m. Since the measurement is affected by a random factor (due to the $R$ matrix and the $\sigma_{UERE}$), the distance error introduced in the outliers oscillates around the 50m value.

Second, an orbit, based on a segment of the LEO orbit, was modeled to simulate trajectory correction maneuvers during thrust phases. The acceleration considered is based on values:

- On 50th second to the 200th second an acceleration is made of intensity of $+0.005\,m/s^2$ along the $x$-axis.
- On 650th second to the 800th second an acceleration is made of intensity of $+0.05\,m/s^2$ along the $z$-axis.

The second value of acceleration is higher than the usual in trajectory correction and higher than the
accelerations referenced on the paper. This value was considered to push the filter to its limit. Notice that this orbit was not calculated using a perturbations model, and the only acceleration mismatch relative to the dynamic model are the ones caused by the maneuvers.

Figure 3.3: Reference orbit used as maneuvers example.
3.2 Filter Implementation

3.2.1 Equations of Motion

In order to determine the receiver’s position in three dimensions \((x_u, y_u, z_u)\) and the receiver’s clock offset \(t_u\), pseudorange measurements are made to \(M\) satellites resulting in the following system of equations:

\[
\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2 + c t_u + \epsilon_i} \quad i \in \{1, ..., M\} \tag{3.1}
\]

where \((x_i, y_i, z_i)\) stand for the \(i\)th satellite’s position in three dimensions, \(\epsilon_i\) denotes the error associated to each measurement \(\rho_i\) due to the errors of ephemeris, thermal noise, ionosphere, etc. These errors are analyzed in detail in Subsection 3.2.4.

These nonlinear equations can be linearized assuming a single pseudorange:

\[
\hat{\rho}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2 + c \hat{t}_u} \quad i \in \{1, ..., M\} \tag{3.2}
\]

where the errors are neglected. Using the approximate receiver’s location \((\hat{x}_u, \hat{y}_u, \hat{z}_u)\) and clock offset’s estimate \(\hat{t}_u\), the approximate pseudorange becomes:

\[
\hat{\rho}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2 + c \hat{t}_u} \quad i \in \{1, ..., M\} \tag{3.3}
\]

Although the receiver’s position and clock offset is unknown, it is considered to consist of an approximate component and an incremental component [Kaplan, 2005]

\[
x_u = \hat{x}_u + \Delta x_u
\]
\[
y_u = \hat{y}_u + \Delta y_u
\]
\[
z_u = \hat{z}_u + \Delta z_u
\]
\[
t_u = \hat{t}_u + \Delta t_u
\]

Therefore, Eq. 3.2 can be expanded about the approximate point and associated predicted receiver clock offset using a Taylor series:

\[
\rho_i = \hat{\rho}_i + \frac{\partial \hat{\rho}_i}{\partial x_u} \Delta x_u + \frac{\partial \hat{\rho}_i}{\partial y_u} \Delta y_u + \frac{\partial \hat{\rho}_i}{\partial z_u} \Delta z_u + \frac{\partial \hat{\rho}_i}{\partial t_u} \Delta t_u + ... \tag{3.4}
\]

The expansion is truncated after the first-order partial derivatives to not consider nonlinear terms. The partial derivatives evaluate as follows:

\[
\frac{\partial \hat{\rho}_i}{\partial x_u} = -\frac{x_i - \hat{x}_u}{r_i}
\]
\[
\frac{\partial \hat{\rho}_i}{\partial y_u} = -\frac{y_i - \hat{y}_u}{r_i}
\]
\[
\frac{\partial \hat{\rho}_i}{\partial z_u} = -\frac{z_i - \hat{z}_u}{r_i} \tag{3.5}
\]
\[
\frac{\partial \hat{\rho}_i}{\partial \Delta c_t} = 1
\]

where

\[
\hat{r}_i = \sqrt{(x_i - \hat{x}_u)^2 + (y_i - \hat{y}_u)^2 + (z_i - \hat{z}_u)^2}
\]  

(3.6)

Substituting Eq. 3.5 into Eq. 3.4 and rearranging the expression, this yields

\[
\hat{\rho}_i - \rho_i = \frac{x_i - \hat{x}_u}{\hat{r}_i} \Delta x_u + \frac{y_i - \hat{y}_u}{\hat{r}_i} \Delta y_u + \frac{z_i - \hat{z}_u}{\hat{r}_i} \Delta z_u - \Delta c t_u
\]

(3.7)

The initial Eq. 3.3 is now linearized with respect to the unknowns. Note that error sources such as measurement noise, propagation delays, among others, have been neglected. The previous equation can be arranged for convenience in the following way:

\[
\Delta \rho_i = a_{xi} \Delta x_u + a_{yi} \Delta y_u + a_{zi} \Delta z_u - \Delta c t_u
\]

(3.8)

where

\[
a_{xi} = \frac{x_i - \hat{x}_u}{\hat{r}_i}
\]

\[
a_{yi} = \frac{y_i - \hat{y}_u}{\hat{r}_i}
\]

\[
a_{zi} = \frac{z_i - \hat{z}_u}{\hat{r}_i}
\]

The vector \((a_{xi}, a_{yi}, a_{zi})\) represents the direction cosines of the unit vector from the spacecraft’s approximate position to the \(i\)th GPS satellite.

These equations can be written in matrix form using these definitions:

\[
\Delta \rho = \begin{bmatrix} \Delta \rho_1 \\ \Delta \rho_2 \\ \vdots \\ \Delta \rho_n \end{bmatrix}, \ H = \begin{bmatrix} a_{x1} & a_{y1} & a_{z1} & -1 \\ a_{x2} & a_{y2} & a_{z2} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{xn} & a_{yn} & a_{zn} & -1 \end{bmatrix}, \ \Delta \nu = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ \Delta c t_u \end{bmatrix}
\]

(3.10)

by which one gets:

\[
\Delta \rho = H \Delta \nu
\]

(3.11)

This linearization will work well as long as the displacement \((\Delta x_u, \Delta y_u, \Delta z_u)\) is within close proximity of the linearization point [Kaplan, 2005]. The acceptable displacement is dictated by the receiver’s accuracy requirements. Indeed, the true receiver-to-GPS satellite measurements are corrupted by independent errors, such as thermal noise, deviation of the satellite path from the reported ephemeris, and others.
3.2.2 Dynamic Model

The filter dynamics is dependent on the adopted model. Since the spacecraft receiver is in motion, the adopted model is a PV-model (Position + Velocity), meaning that the state vector has eight elements: three elements for the position, three elements for the velocity and the last two for the clock characterization (as detailed in Subsection 3.2.2.A).

In the PV-model each coordinate can be modeled as an integrated brownian motion, represented in Fig. 3.4.

Figure 3.4: PV model state space. (Brown, 1997).

This model can be interpreted into the matricial form as

\[
\begin{bmatrix}
  x_{1,k+1} \\
  x_{2,k+1} \\
  x_{3,k+1} \\
  x_{4,k+1} \\
  x_{5,k+1} \\
  x_{6,k+1} \\
  x_{7,k+1} \\
  x_{8,k+1}
\end{bmatrix}
= \begin{bmatrix}
  1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\
  0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_{1,k} \\
  x_{2,k} \\
  x_{3,k} \\
  x_{4,k} \\
  x_{5,k} \\
  x_{6,k} \\
  x_{7,k} \\
  x_{8,k}
\end{bmatrix}
+ \begin{bmatrix}
  u_{1,k} \\
  u_{2,k} \\
  u_{3,k} \\
  u_{4,k} \\
  u_{5,k} \\
  u_{6,k} \\
  u_{7,k} \\
  u_{8,k}
\end{bmatrix}
\]  

(3.12)

assuming \( x_1, x_2 \) as the position and the velocity of the receiver, respectively. The covariance noise matrix is

\[
Q_k = E\left( [u_{1,k}u_{2,k}]^T [u_{1,k}u_{2,k}] \right) = q_v \Delta t \begin{bmatrix}
  \Delta t^2 & 0 \\
  0 & \frac{\Delta t}{2}
\end{bmatrix}
\]  

(3.13)

Following Eq. 2.27, the integration of the nonlinear state yields a correction on the velocity estimate state, given that the force is known, the acceleration follows

\[
a_{i,k} = \frac{GM}{||r||^3} r , \quad i = \{2, 4, 6\}
\]  

(3.15)

where \( r \) is the position vector of the spacecraft estimated by the filter and \( ||r|| \) is its norm. The acceleration is computed along each of the three axes. From Eq. 3.13 the final covariance noise matrix is presented in given by
Q

\[
Q_k = \begin{bmatrix}
q_v \Delta t^2 & q_v \Delta t & q_v \\
q_v \Delta t^2 & q_v \Delta t & q_v \\
q_v \Delta t^2 & q_v \Delta t & q_v
\end{bmatrix}
\]

(3.16)

The quantity \( q_v \) can be estimated using Eq. 3.12 (Nunes, 2013)

\[
q_v = E\{u^2_{2,k}\} \geq \Delta t^2 a_{res}^2
\]

(3.17)

where \( a_{res} \) is an estimate of the residual acceleration not modeled by the dynamic model. This includes the acceleration produced by all external perturbations as firing thrusters on maneuvers, Earth’s shape, radiation, drag, and others.

### 3.2.2. A Receiver’s clock state space

The GPS receiver clock introduces time-varying timing error that is converted into error ranging error affecting measurements made of all GPS satellite’s. Since the measurements are made simultaneously, the clock error is the same on all measurements. Being so, there are advantages in modeling the clock. A suitable clock model is a two-state random process model (Brown, 1997). This way it is possible to model the clock assuming that both the oscillator phase and frequency are described by a random walk over time. In Fig. 3.5 the block diagram is presented.

![Figure 3.5: Receiver clock state space](edited from Brown, 1997)

Both white noises represented above are independent, zero-mean and characterized by the following covariance matrix (Nunes, 2013)

\[
Q_u = \begin{bmatrix}
q_\phi & 0 \\
0 & q_f
\end{bmatrix}
\]

(3.18)

where \( q_f \) and \( q_\phi \) are, respectively, the spectral power densities of the signals \( u_f \) and \( u_\phi \) (input signals in Fig. 3.5). The discrete-time model corresponds to (Nunes, 2013)

\[
\begin{bmatrix}
x_\phi,k+1 \\
x_f,k+1
\end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_\phi,k \\
x_f,k
\end{bmatrix} + \begin{bmatrix} u_\phi,k \\
u_f,k
\end{bmatrix}
\]

(3.19)

The covariance matrix associated is
\[ Q_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) Q_u \Phi^T(t_{k+1}, \tau) d\tau = \left[ q_0 \Delta t + \frac{q_f \Delta t^3}{3} \frac{q_f \Delta t^2}{2} \right] \] (3.20)

The variances of matrix in Eq. 3.18 are derived in [Brown, 1997] and can be approximated as

\[ q_0 \approx \frac{h_0}{2} \] (3.21)

\[ q_f \approx 2\pi^2 h_2 \]

The used values follow the description of the receiver that uses a TCXO oscillator. Later on in Section 3.2.4 the receiver is fully characterized. Table 3.1 presents the values used and how they relate with other commonly used oscillators.

**Table 3.1:** Parameters for the Allan variance of several oscillators [Winkel, 2003]

<table>
<thead>
<tr>
<th>Oscillator</th>
<th>White freq. noise ( (h_0) )</th>
<th>Flicker ( (h_{-1}) )</th>
<th>Integrated freq. noise ( (h_{-2}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard quartz</td>
<td>( 2 \times 10^{-19} ) s</td>
<td>( 7 \times 10^{-21} )</td>
<td>( 2 \times 10^{-20} ) Hz</td>
</tr>
<tr>
<td>TCXO</td>
<td>( 1 \times 10^{-21} ) s</td>
<td>( 1 \times 10^{-20} )</td>
<td>( 2 \times 10^{-20} ) Hz</td>
</tr>
<tr>
<td>OCXO</td>
<td>( 8 \times 10^{-20} ) s</td>
<td>( 2 \times 10^{-21} )</td>
<td>( 4 \times 10^{-23} ) Hz</td>
</tr>
<tr>
<td>Rubidium</td>
<td>( 1 \times 10^{-23} ) s</td>
<td>( 1 \times 10^{-22} )</td>
<td>( 1.3 \times 10^{-26} ) Hz</td>
</tr>
</tbody>
</table>

TCXO - Temperature Compensated Crystal Oscillator  
OCXO - Overnized Crystal Oscillator (temperature controlled)

### 3.2.3 Observation Model

The observation model follows a nonlinear and time-dependent equation:

\[ y_k = h(x(k)) + v_k \] (3.22)

where the vector \( z_k = [\rho_{1,k}, \rho_{2,k}, ..., \rho_{n,k}]^T \) with \( n \geq 4 \) is the pseudorange vector. The matrix \( h(.) \) is described by

\[ h_k = \begin{bmatrix} \sqrt{(\tilde{x}_1 - x_a)^2 + (\tilde{y}_1 - x_b)^2 + (\tilde{z}_1 - x_c)^2 + x_d} \\ \vdots \\ \sqrt{(\tilde{x}_n - x_a)^2 + (\tilde{y}_n - x_b)^2 + (\tilde{z}_n - x_c)^2 + x_d} \end{bmatrix} \] (3.23)

where \( (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i) \) are the coordinates of the \( i \)th satellite and \( x_a, x_b, x_c, x_d \) are components of the state vector of \( x_u, y_u, z_u \) and \( c \Delta t \).

The noise is assumed to be coordinate-independent. Therefore, the observation covariance noise matrix is a diagonal matrix in the form of:

\[ R_k = \begin{bmatrix} \sigma_{1, UERE}^2 & 0 \\ 0 & \sigma_{2, UERE}^2 \\ \vdots & \vdots \\ 0 & \sigma_{n, UERE}^2 \end{bmatrix} \] (3.24)

In Subsection 3.2.4 the noise and \( \sigma_{n, UERE} \) are characterized in detail.

As previously seen in Eq. 2.26, \( H_k \) matrix represents the Jacobian matrix of \( h(.) \), and matricially this yields:
where $n$ is the number of satellites observed, and $[\hat{x}_u, \hat{y}_u, \hat{z}_u] = [\hat{x}_{1|k+1}, \hat{x}_{3|k+1}, \hat{x}_{5|k+1}]$. For the PV-model, this results in the following $H_k$ matrix.

$$H_k = \begin{bmatrix}
 a_{x1} & 0 & a_{y1} & 0 & a_{z1} & 0 & -1 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 a_{xn} & 0 & a_{yn} & 0 & a_{zn} & 0 & -1 & 0 
\end{bmatrix}$$

(3.26)

where

$$a_{xi} = \frac{\tilde{x}_i - \hat{x}_u}{\hat{r}_i}, \quad a_{yi} = \frac{\tilde{y}_i - \hat{y}_u}{\hat{r}_i}, \quad a_{zi} = \frac{\tilde{z}_i - \hat{z}_u}{\hat{r}_i}$$

(3.27)

in which the $\hat{r}_i = \sqrt{((\tilde{x}_i - \hat{x}_u)^2 + (\tilde{y}_i - \hat{y}_u)^2 + (\tilde{z}_i - \hat{z}_u)^2)}$.

### 3.2.4 Noise Generation

When dealing with GPS measurement errors, the tracking thresholds assume a major role, since the receiver loses lock when the error exceeds a given boundary. Monte Carlo simulations of the GPS receiver under specific conditions enable the determination of the true tracking performance [Ward, 1997], mainly due to the nonlinear tracking loops of the code and carrier.

Despite this, there are general rules based on closed form equations that approximate the measurement errors of the tracking loops. Several sources of measurement errors are in each type of tracking loop. However, it is sufficient for rule-of-thumb tracking thresholds to analyze only the dominant error sources [Kaplan, 2005].

In the situation of this work, all communications are made from the GPS to the spacecraft receiver. Assuming no multipath or other interferences, the dominant sources of range error in the GPS receiver code tracking loop (DLL) are thermal noise range error jitter and dynamic stress error [Kaplan, 2005].

A general expression, on the given environment, for thermal noise code tracking jitter for a noncoherent DLL discriminator is proposed in [Betz, 2009]. For BPSK modulations, when using a noncoherent early-late power DLL discriminator, the thermal noise tracking jitter can be approximated by [Kaplan, 2005]:

$$\sigma_{tDLL} \approx \sqrt{\frac{B_n}{2C/N_0}D \left[ 1 + \frac{2}{T C/N_0 (2 - D)} \right]} \text{[chips]}, \quad D \geq \frac{\pi R_c}{B_{fe}}$$

(3.28)

where:

- $B_{fe}$ - double-sided front-end bandwidth (Hz)
- $R_c$ - spreading code chip rate (Hz)
- $D$ - Early-Late spacing: 0.25 chips
- $T$ - Integration time: 0.004 s
- $B_n$ - DLL bandwidth: 0.5 Hz
- $C/N_0$ - carrier to noise power expressed as a ratio, in (dB-Hz): variable

Given that the variation of $C/N_0$ may be a relevant contributor for the error, Fig. 3.6 illustrates the impact of the parameter on $\sigma_{tDLL}$.

![Figure 3.6: Evolution of $\sigma_{tDLL}$ with a linear $C/N_0$.](image)

Note that the values of $C/N_0$ are in linear units. The x-axis is represented from $\sim [2; 100] \text{Hz}$, equivalent to $[10; 55] \text{dB - Hz}$.

Other expressions for other environments and receiver characteristics can be found in [Kaplan, 2005] and [Betz, 2009].

The carrier-to-noise power ratio, $C/N_0$, is an important factor in many GPS receiver performance measures. It is computed as the ratio of recovered power, $C$, (in W) from the signal to the noise density $N_0$ (in W/Hz). In a broad sense, if $C/N_0$ increases, the standard deviation decreases, according to Eq. 3.28.

The right-hand side term in brackets is the squaring loss. This means that increasing $T$ the squaring loss in DLLs is reduced. This term becomes unitary if using a coherent DLL discriminator (no squaring loss). Also, increase in the integration time $T$ results in a lower $C/N_0$ threshold. Other conclusions can be drawn from the expression: reducing $D$, the correlator spacing, also implies a reduction on the DLL jitter at the cost of increased code tracking sensitivity.

The calculation of the $C/N_0$ parameter is non-trivial. Since the spacecraft has a certain orientation, the receiver radiation pattern orientation is dependent of the spacecraft orientation. Also, the distance of the spacecraft to different GPS satellites is different and variable. The solution for this problem is out of the scope of this thesis, therefore the values for $C/N_0$ were calculated using a closed-source software simulator developed by Deimos Engenharia Granada GNSS Blockset.
This simulation used as input the position and velocity of the spacecraft, as well as the position and velocity of the GPS constellation throughout an entire period of the receiver’s orbit. Doing so, for each timestep, a certain GPS satellite would have a different value of $C/N_0$, or “- inf” if not visible.

The characteristics of the simulated receiver are:

- Signal GPS L1 C/A
- Receiver Antenna Temperature: $170 \, K$
- Receiver Implementation Losses: $2 \, dB$
- Receiver Front-End Noise Figure: $1.5 \, dB$
- Receiver antenna pattern standard for a LEO satellite (zenith pointing). For a detailed characterization, Fig 3.7 represents the boresight angle versus gain.
- GNSS satellites antenna pattern (EIRP), according to literature [Marquis, 2014]

![Figure 3.7: Receiver antenna pattern characterization.](image)

The output result for the $\sigma_{IDLL}$ parameter in Eq. 3.28 is in chips, and for the desired conversion to meters, one needs to correct the dimensions:

$$\sigma_{IDLL} \, [m] = \sigma_{IDLL} \, [\text{chips}] \times \text{speed of light} \times \frac{1 \times 10^{-3}}{1023}$$  

with $10^{-3}/1023s$ standing for the chip duration.

The thermal noise is the most significant source of error and accounts for geometric orientation of the receiver, geometric orientation of the satellite, distance of the GPS to the receiver, among others. This allows to weight the pseudorange measurements for different GPS satellites used.

There are also other non-variable sources of error that affect all satellites and that were accounted:
The ephemeris data is transmitted every 30 seconds, but the information may be up to two hours old. Also uncertainties in solar radiation pressure model, gravity field model and others, affect indirectly the ephemeris errors (Colombo, 1986). For terrestrial receivers there are solutions for this problem, such as Assisted GPS.

The satellite’s atomic clocks are under the influence of noise and clock drift errors. The navigation message estimates the correction for these errors, but they are based on observations and may not indicate the clock’s real current state.

However less significant, ephemeris and clock errors were also accounted in this work, as well as Group L1 delay. These three smaller effects were modeled in addition to the major one - thermal noise.

<table>
<thead>
<tr>
<th>Noise Source</th>
<th>Error GPS 1 σ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal Noise (σ_{DLL})</td>
<td>typically [0.2;1.5]</td>
</tr>
<tr>
<td>GNSS Ephemeris (σ_{eph})</td>
<td>0.8</td>
</tr>
<tr>
<td>Group L1 delay (σ_{L1})</td>
<td>0.3</td>
</tr>
<tr>
<td>GNSS Satellite Clock (σ_{clock})</td>
<td>0.6</td>
</tr>
</tbody>
</table>

This noise characterization allows the simulation to be more realistic, however, it affects the different implement algorithms equally and a more sophisticated noise model would not, in theory, affect a particular algorithm nor its relative comparison.

The final measurement error with all the contributions becomes:

\[ \sigma^2_{UERE} = \sigma^2_{tDLL} + \sigma^2_{eph} + \sigma^2_{L1} + \sigma^2_{clock} \ [m^2] \] (3.30)

### 3.2.5 Parameters tuning

#### 3.2.5.A Parameters of the Robust Extended Kalman Filter

The \( \gamma \) parameter follows the previously seen Eq. 2.39. However, it is not possible to model accurately the external disturbances, and the following stability condition was used in favour of the previous one:

\[ \gamma > \left\{ \max \left[ \text{eig} \ (P_{k|k-1}) \right] \right\}^{\frac{1}{2}} \] (3.31)

Doing so, \( \gamma \) becomes a time-varying parameter, and the condition \( \Sigma_{k|k-1} > 0 \) is always satisfied. When disturbances are rather large, it is difficult to choose an appropriate \( \gamma \) such that \( \Sigma_{k|k-1} > \Sigma_{k|k-1} \) and the previous condition are verified. On the next chapter a more detailed analysis will be made on this topic.

For some applications a fixed \( \gamma \) that matches both stability conditions is possible to find, which requires less computational effort.

#### 3.2.5.B Parameters of the Adaptive Robust Extended Kalman Filter

The main difference from REKF is that the prediction error covariance matrix is supposed to be calculated by Eq. 2.34 instead of Eq. 2.33. The design of the parameter \( \lambda_k \) is of most importance to

37
control the accuracy and stability of the filter. For an appropriate $\lambda_k$, this parameter should be tuned according to the trace of the innovation covariance matrix $\tilde{P}_{y,k}$ such as (Xiong, 2009)

$$
\lambda_k = \frac{\text{trace}(\tilde{P}_{y,k})}{\text{trace}(\tilde{P}_{\tilde{y},k})}
$$

(3.32)

Another problem arises when computing the auxiliary matrix $\tilde{P}_{y,k}$ from Eq. 2.55. Since the number of satellites varies for each timestep, the dimension of the innovations vector $\tilde{y}_k$ also varies iteratively, implying that matrix $\tilde{P}_{y,k-1}$ and $\tilde{y}_k \tilde{y}_k^T$ may not have the same size. For both the EKF and REKF this was not an issue, since the Kalman Gain computation made all dimensions agree.

The solution to this problem was to maintain the size of the covariance matrix fixed to the maximum number of visible satellites on the entire run (for instance, if the number of visible satellites was between 8 and 19 satellites, the covariance matrix would be a $[19 \times 19]$ matrix). A value of “$\infty$” was given to the covariance matrix for the entries that do not have any visible satellite. Doing so, the covariance error matrix is constant in size throughout the run and the weight of the non-visible satellites is null. This adaptation is not optimal, since it requires an extra computational effort for non-visible satellites. Also, according to the orbit used, the maximum number of satellites should be determined a priori, in order to use the maximum possible information available.

Another alternative strategy would be to consider, a prior, a minimum number of visible satellites and always use that number on the algorithm. When the number of visible satellites was larger than that fixed number, satellites would be chosen based on a given criteria, for instance, using the minimization of GDOP.
Testing and Results

Contents

4.1 Simulation Environment .................................................. 40
4.2 Simulation Results .......................................................... 42
This chapter discusses the simulation environment, in Section 4.1, where simulation characteristics of the reference orbit used, a description of the $C/N_0$ and other characteristics, are presented.

The simulation results are covered in Section 4.2. First the EKF results for two values of parameter $q_v$ are presented and discussed in Section 4.2.1, followed by the REKF - Section 4.2.2 - results, where an analysis on $\gamma$ parameter is made. Next the AREKF and the parameter $\alpha$ are studied in Section 4.2.3 with special attention given to the filter output for the variation of $\alpha$.

In Section 4.2.4 a comparison between algorithms is made for the LEO orbit, and then in Section 4.2.5 the comparison is made for other two environments: measurement outlier - 4.2.5.A - and orbital maneuvers - 4.2.5.B.

### 4.1 Simulation Environment

All simulations and analysis were made using MATLAB R2015a version.

The modeled noise described in Section 3.2.4 considering a simplified perturbations model was added to the reference orbit generated with the almanac. The result for the LEO orbit is shown in Fig. 4.1.

![Figure 4.1: Reference orbit with modeled noise for a LEO orbit in ECEF coordinates (T = 5923 secs).](image)

The $C/N_0$ values used throughout a period of a LEO orbit are presented in Fig. 4.2. The plot corresponds to the GPS constellation of 32 satellites and it was generated as described in Section 3.2.4.
As expected, besides the overall intensity of the signal, it is possible to see that the availability of each satellite also varies with the time, being typically between $[8;16]$ for a LEO orbit.

Throughout the analysis there are also other simulation characteristics that were maintained fixed such as the represented in Table 4.1.
Table 4.1: Environment Characteristics.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter cycle rate</td>
<td>1 [Hz]</td>
</tr>
<tr>
<td>Speed of light</td>
<td>299792458 [m/s]</td>
</tr>
<tr>
<td>Gravitational Constant</td>
<td>$6.674287 \times 10^{-11} [Nm/kg^2]$</td>
</tr>
<tr>
<td>Earth Mass</td>
<td>$5.97219 \times 10^{24} [kg]$</td>
</tr>
</tbody>
</table>

Some GPS receivers provide the 1 Pulse-Per-Second (PPS) signal, and this was the case assumed on this work. Since the filter rate and the GPS receiver output rate are considered the same, no synchronization was required.

4.1.1 Filtering Tuning

The initial parameters of the filters have to be previously defined for correct estimation. The initial state estimation position is defined as follows:

$$\hat{x}_{pos} \sim \mathcal{N}(x_{pos} ; 10) [m]$$  \hspace{1cm} (4.1)

As for the initial state velocity, a smaller variance was given:

$$\hat{x}_{vel} \sim \mathcal{N} \left( \frac{x_{k+1} - x_k}{\Delta step} ; 5 \right) [m/s]$$  \hspace{1cm} (4.2)

The initial condition for the error covariance matrix is supposed to be large for unknown initial state error or small if the initial state is known. An intermediate level of confidence was assumed, and as such, the $P$ matrix becomes:

$$P_{0|0} = \text{diag}(50, 50, 50, 20, 20, 20, 2, 2)$$  \hspace{1cm} (4.3)

The measurements noise covariance matrix $R$ was modeled carefully, varying dynamically with $C/N_0$, and therefore, no tuning was made. The only tuned parameter was $q_v$, described in Section 3.2.2 part of $Q$ matrix and estimated initially using Eq. 3.13. A more trial and error approach to define the $q_v$ parameter was carried out, adjusting its value manually with the help of simulations.

The optimal value obtained for $q_v$ was $2 (m/s)^2$. The lower the $q_v$, the lower the process noise covariance matrix $Q$ becomes. Since the reference orbit used was calculated using a perturbation model (SGP4) that is not considered on the filter dynamic, there is a non-modeled acceleration. This difference is even more relevant as we decrease the value of $q_v$, since the process noise is decreasing altogether. For a value of $0.001 (m/s)^2$, for instance, the disturbances were very relevant. These values are used as baseline on the following analysis as an example of presence and absence of non-modeled accelerations.

4.2 Simulation Results

The main analysis will be made for a LEO period $(T = 5932 s)$ where all the three methods converged. For every value of $q_v$ tested, none of the three algorithms diverged. The dynamic model, as well as the observation model for the three methods were the same.
The error used on the analysis was the Root Mean Square Error (RMSE) which follows Eq. 4.4:

$$
\epsilon_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_{pos}^i - \hat{x}_{pos}^i)^2 + (y_{pos}^i - \hat{y}_{pos}^i)^2 + (z_{pos}^i - \hat{z}_{pos}^i)^2} \ [m]
$$

where $N$ is the total number of points used. Also, an error for each axis is defined according to Eq. 4.5:

$$
\Delta x = x_1 - \hat{x}_1 \ [m] \\
\Delta y = x_3 - \hat{x}_3 \ [m] \\
\Delta z = x_5 - \hat{x}_5 \ [m]
$$

The square root of the Square Root of the Trace (Diagonal) of the Covariance Matrix (SDCM) follows Eq. 4.6 and is useful for the validation of the algorithms.

$$
SDCM = \pm \sqrt{\text{trace}(P)}
$$

4.2.1 Extended Kalman Filter

For the EKF analysis, the output error is presented for the two values of the parameter $q_v$.

![Graph of EKF error for $q_v = 2 \ (m/s)^2$.](image)

Figure 4.4: EKF error for $q_v = 2 \ (m/s)^2$. 
On the run for the value of \( q_v = 2 \) (m/s)\(^2\), Fig. 4.4, the error is always smaller than the square root of the trace of the covariance matrix. The filter converged rapidly and the error is rather constant on the entire run.

On the run for the smaller value of \( q_v \), Fig. 4.5, the filter never diverges but exceeds (in module), mostly in the \( x \)- and \( y \)-axis, the square root of the covariance matrix. This is due to the acceleration mismodeling being more noticeable under these conditions.

The results for the RMSE are presented in Table 4.2.

<table>
<thead>
<tr>
<th>( q_v )</th>
<th>RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 2 ) (m/s)(^2)</td>
<td>1.0732 ± 0.0087</td>
</tr>
<tr>
<td>( 0.001 ) (m/s)(^2)</td>
<td>13.4132 ± 0.0106</td>
</tr>
</tbody>
</table>

The innovations throughout a period of a LEO orbit of a EKF can be seen in Appendix B.

### 4.2.2 Robust Extended Kalman Filter

For the REKF analysis, the output error is presented for the two values of the parameter \( q_v \) and a variable \( \gamma \) parameter.

On Fig. 4.6 a typical value for \( \gamma \) was defined \((\gamma = 5 \sqrt{\max(eig(P))})\). The RMSE is always smaller (in module) than the SDCM and the output is similar to the EKF for the same value of \( q_v \).

On Fig. 4.7 the same value for \( \gamma \) was defined \((\gamma = 5 \sqrt{\max(eig(P))})\). The RMSE has an identical behaviour of the EKF for the same value of \( q_v \), but with a lower overall error value.

As seen in Subsection 2.1.4.A a stability condition for \( \gamma \) demands this parameter to be greater than \( \sqrt{\max(eig(P))} \). Being so, several values for \( \gamma \) were tested, in order to cover its entire domain and extract conclusions.
Figure 4.6: REKF error for $\gamma = 5, q_v = 2 \, (m/s)^2$.

Figure 4.7: REKF error for $\gamma = 5, q_v = 0.001 \, (m/s)^2$.
The values for the REKF - and the EKF as comparison - are presented in Table 4.3 and 4.4.

Table 4.3: REKF RMSE results for $q_v = 2 (m/s)^2$ for 20 runs

<table>
<thead>
<tr>
<th>$\gamma = \sqrt{\text{max}(\text{eig}(P))}$</th>
<th>REKF RMSE [m]</th>
<th>EKF RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00005</td>
<td>1.2165 ± 0.0085</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>1.1218 ± 0.0074</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.0825 ± 0.0090</td>
<td>1.0732 ± 0.0087</td>
</tr>
<tr>
<td>50</td>
<td>1.0736 ± 0.0091</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>1.0713 ± 0.0079</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: REKF RMSE results for $q_v = 0.001 (m/s)^2$ for 20 runs

<table>
<thead>
<tr>
<th>$\gamma = \sqrt{\text{max}(\text{eig}(P))}$</th>
<th>REKF RMSE [m]</th>
<th>EKF RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00005</td>
<td>10.4564 ± 0.0074</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>10.6700 ± 0.0080</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11.8738 ± 0.0068</td>
<td>13.4132 ± 0.0106</td>
</tr>
<tr>
<td>50</td>
<td>13.4113 ± 0.0124</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>13.4184 ± 0.0105</td>
<td></td>
</tr>
</tbody>
</table>

For the optimal $q_v (2 (m/s)^2)$, the REKF RMSE is always larger than the EKF’s and the maximum difference (greater RMSE) is for the lower possible value of $\gamma$. When $\gamma$ increases, the RMSE of the REKF decreases and tends to the EKF. The values for the two larger $\gamma$’s are already close to the limit of $\gamma \to \infty$, where the REKF reverts to the EKF (according to Eq. 2.34). This tendency was already pointed on the Fundamentals chapter, in Section 2.1.4 and the results are according to the theory.

For the visible perturbations case, the same tendency of REKF to EKF is also present, but in this case, the REKF RMSE is predominantly lower than the EKF.

A further study of the variation of the RMSE with both $\gamma$ and $q_v$ was performed and is presented in Fig. 4.8.

Figure 4.8: REKF RMSE normalized with the variation of $\gamma$ and $q_v$ for a LEO orbit.
where the RMSE normalization follows Eq. 4.7

\[ \text{RMSE}_{\text{NORM}} = \frac{\text{REKF}_{\text{RMSE}} - \text{EKF}_{\text{RMSE}}}{\text{EKF}_{\text{RMSE}}} + 1 \] (4.7)

This illustrates that with the decrease of \( q_v \) (increase of non-modeled accelerations), implementing the REKF becomes an advantage. There is a tipping point for a value of \( q_v \) that the REKF outperforms the EKF. Depending on the scenario and the intensity of the non-modeled acceleration, the REKF may outperform the EKF in terms of RMSE.

Note that for each of the points plotted only one run was performed, and therefore the plotted lines lack in smoothness, since the values represented have not converged to its true value.

Notice that in this case the acceleration is constant; this analysis would not be possible for a maneuver, where the intensity of the acceleration would be variable and this tipping point analysis would not be conclusive.

It was previously seen, in Section 2.1.4, that when lowering \( \gamma \) the robustness of the filter would increase, and that would come at the cost of RMSE performance - due to the Kalman Gain computation (see Eq. 2.34 and Eq. 2.36). This is evident on the results of the optimal value of \( q_v \), in which the EKF is the desirable algorithm to implement. The robust characteristic of the filter is not required, and one only loses optimality. However, when the perturbations are evident (\( q_v = 0.001 \text{ (m/s)}^2 \)), the REKF is the desirable algorithm to implement, according to a RMSE based analysis.

### 4.2.3 Adaptive Robust Extended Kalman Filter

As for the AREKF analysis, the output error is presented in Figs. 4.9 and 4.10 for the two values of the parameter \( q_v \), a fixed \( \gamma \) parameter of \( 5 \sqrt{\text{max(eig(P))}} \) and a fixed \( \alpha \) of 0.6. As seen in Subsection 2.1.5.A a stability condition demands that \( \alpha \) to be always positive. Being so, several positive values for \( \alpha \) were tested, in order to cover its domain and extract conclusions.

In Fig. 4.9 the RMSE is always smaller (in module) than the SDCM and the output is similar to the EKF and REKF for the same value of \( q_v \).

In Fig. 4.7 the RMSE has an identical behaviour of the REKF and EKF for the same value of \( q_v \), but with a lower overall error value.

As seen in Section 2.1.5, the main difference between AREKF and REKF is that the prediction error covariance matrix is calculated by Eq. 2.54 instead of Eq. 2.34. Using this equation it is possible to predict that for smaller values of \( \alpha \) the filter tends to an EKF filter, and for larger values of \( \alpha \), the filter reverts to a REKF-alike (since matrix \( L \) is not exactly the same). Since the \( \alpha \) parameter is what makes the switch of the 'mode' of the filter, its tuning is of most importance to control the accuracy of the filter.

The values for the AREKF - and the REKF/EKF as comparison - with a varying \( \alpha \) are presented in Table 4.5 and Table 4.6.
Figure 4.9: AREKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $\alpha = 0.6$ and $q_v = 2 \,(m/s)^2$.

Figure 4.10: AREKF error for $\gamma = 5 \sqrt{\max(eig(P))}$, $\alpha = 0.6$ and $q_v = 0.001 \,(m/s)^2$. 
Figure 4.11: AREKF RMSE evolution with $\alpha$, for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 0.001 (m/s)^2$.

Table 4.5: AREKF RMSE results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 2 (m/s)^2$ for 20 runs

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>AREKF RMSE [m]</th>
<th>EKF RMSE [m]</th>
<th>REKF RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0001</td>
<td>1.0733 ± 0.0070</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>1.0757 ± 0.0085</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.1529 ± 0.0084</td>
<td>1.0732 ± 0.0087</td>
<td>1.2165 ± 0.0085</td>
</tr>
<tr>
<td>50</td>
<td>1.1538 ± 0.0098</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>1.2071 ± 0.0081</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6: AREKF RMSE results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 0.001 (m/s)^2$ for 20 runs

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>AREKF RMSE [m]</th>
<th>EKF RMSE [m]</th>
<th>REKF RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0001</td>
<td>13.8596 ± 0.0118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>13.3263 ± 0.0120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>12.5141 ± 0.0103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>9.5454 ± 0.0099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>8.7786 ± 0.0083</td>
<td>13.4132 ± 0.0106</td>
<td>10.4564 ± 0.0074</td>
</tr>
<tr>
<td>1</td>
<td>9.1265 ± 0.0080</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10.7301 ± 0.0077</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>10.7941 ± 0.0106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>10.8699 ± 0.0085</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The entire results for the AREKF for $q_v = 0.001 (m/s)^2$ (in an extensive table) are presented in Appendix B in Section B.2.

In both cases of $q_v$, the limits of $\alpha \to 0$ and $\alpha \to +\infty$, yield similar results to the EKF and REKF, respectively. However, unlike the $\gamma$ parameter, $\alpha$ variation does not have a monotone tendency on the RMSE. It is clear that there is a decrescent tendency, and then a crescent behaviour. A more detailed study was made on this interval. Figure 4.11 shows the plot with a zoom on this area.
The plot illustrates the decreasing and increasing tendency. The points were plotted with error bars using the standard deviation of the 20 runs made for each data point. A value of \( \alpha \) that minimizes the RMSE for this specific set of \( \gamma \) and \( q_v \) parameters is possible to extract. In a different environment the value of \( \alpha \) that minimizes the RMSE is different. This happens since the non-modeled acceleration is not so visible (for a higher \( q_v \), for instance), and the best \( \alpha \) that minimizes the RMSE is higher. This way the AREKF will run more on the EKF mode, since the threshold is higher.

### 4.2.4 Algorithms Comparison

A performance comparison on the RMSE was made. A worst case scenario possible of \( q_v \) was simulated (low \( q_v \)) and the error norm of the three algorithms is compared in Fig. 4.12.

![Figure 4.12](image)

*Figure 4.12:* All three algorithms error norm evolution, for \( \gamma = 3\sqrt{\max(eig(P))} \), \( \alpha = 0.7 \) and \( q_v = 0.001 \text{ (m/s)}^2 \).

The AREKF outperforms both competitors, being the EKF outperformed by all its competitors.

As previously seen in Section 2.1.5, for a lower \( q_v \), the filter innovation \( \tilde{y} \) will be enlarged due to the unknown accelerations, and the trace of matrix \( P_{y,k} \) will increase, leading the filter to work on the REKF-alike mode. When innovations are lower, the filter switches to an EKF filter, increasing the RMSE performance. This means that the AREKF autonomously, depending on the innovations, switches from a standard EKF to a REKF-alike filter, being able to reduce the RMSE.

Fig. 4.13 compares the RMSE between the three algorithms.
Both limits of the AREKF are emphasized in this plot, as well as a first decrease in RMSE with the increase of $\alpha$ and then the increase of RMSE. Analyzing the complete table in Appendix B, it is true that for large values of $\alpha$, the RMSE of the AREKF ($10.8699 \pm 0.0085$ m) is a little higher than the RMSE of the REKF ($10.4564 \pm 0.0074$ m). This is due to the difference in the definition of matrix $L_t$ in both methods. However, this degradation of performance only occurs for a wrongfully tuned $\alpha$ of several orders of magnitude.

A comparison in terms of computational times is presented in Table 4.7.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time per run [s]</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>2.768 ± 0.040</td>
<td>100</td>
</tr>
<tr>
<td>REKF</td>
<td>4.344 ± 0.204</td>
<td>157</td>
</tr>
<tr>
<td>AREKF</td>
<td>5.121 ± 0.276</td>
<td>185</td>
</tr>
</tbody>
</table>

The EKF is the fastest algorithm, followed by the REKF and the AREKF. However, no optimization was performed. The difference in the computational times is due to the increase of calculations needed for both the REKF and the AREKF.

4.2.5 Other Environments

4.2.5.A Measurement Outliers

Some other environments were tested besides the LEO standard orbit. First, for a LEO orbit, after the filter converged, outliers in the measurements were introduced. These outliers simulate real instrument errors that occasionally occur in GNSS measurements (e.g., pseudoranges and Doppler).

A run of 300 seconds was made and two consecutive measurements, at $t=150$ s and $t=151$ s, were multiplied by a factor of $(1 + 2 \times 10^{-6})$, corresponding to an error measurement of around 50 meters. The filters RMSE output is represented in Fig. 4.14.

The AREKF line is predominantly under the orange (REKF) line, which is predominantly below the EKF line. The respective RMSE for these runs is presented in Table 4.8.
Figure 4.14: The three algorithms RMSE with outliers for $t = 150s$ and $t = 151s$ [$\alpha = 0.7, \gamma = 3\sqrt{\text{max}(\text{eig}(P))}, q_v = 2$ and 300 secs run].

Table 4.8: All three algorithms RMSE with outliers for $t = 150$ and $t = 151$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RMSE [m]</th>
<th>Max. Peak [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>3.397</td>
<td>48.413</td>
</tr>
<tr>
<td>REKF</td>
<td>1.891</td>
<td>26.103</td>
</tr>
<tr>
<td>AREKF</td>
<td>1.520</td>
<td>17.783</td>
</tr>
</tbody>
</table>

The measurement outlier makes the innovation enlarge abruptly, since the measurement corresponds to a position that is not near the previous estimation. This can be interpreted as an external disturbance and the AREKF outperforms both REKF and the EKF. This is another example where the AREKF, in a RMSE-only based analysis, outperforms both the REKF and the EKF.

However, a RMSE analysis, in this case, may hide the true performance of the AREKF: since $N$ is large, the RMSE of the three methods shows small variation. On the exact instant where the outlier occurs the AREKF outperforms both REKF and EKF by $\sim 46\%$ and $\sim 172\%$ respectively.

4.2.5.B Orbital Maneuvers

Orbital maneuver, as described in Section 3.1.2, was also tested. The acceleration on direction x-axis around $t = 50s$ of intensity $0.005 \text{ (m/s)}^2$ is not noticeable, however the acceleration on the direction z-axis around the 650th second of intensity $0.05 \text{ (m/s)}^2$ is notable.

The error on the different axis in Fig. 4.15 illustrates the thrust accelerations with the EKF algorithm. Recall that this reference orbit was not calculated using perturbations model. Being so, the only non-modeled acceleration is the described above. This was made to isolate the accelerations caused by the maneuvers.

The value of parameter $q_v$ was fine tuned empirically so that the non-modeled acceleration was visible, $\gamma$ and $\alpha$ were set to typical values. The comparison between the three algorithms is represented in Fig. 4.16.
Figure 4.15: EKF error for orbital maneuvers with $q_v = 0.0001 \ (m/s)^2$.

Figure 4.16: All three algorithms error norm with orbital maneuvers $[\alpha = 0.7, \gamma = 3\sqrt{\max\{\text{eig}(P)\)}, q_v = 0.0001, 1000 \text{ secs run}]$. 
Once again the AREKF line is predominantly under the REKF line, which is predominantly below the EKF line. The respective RMSE for these runs is presented in Table 4.9. The RMSE was calculated on the interval where the maneuver was more visible so that the comparison could be more accurate.

**Table 4.9:** All three algorithms RMSE with orbital maneuvers.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>RMSE [650; 850] [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKF</td>
<td>4.7819</td>
</tr>
<tr>
<td>REKF</td>
<td>2.8630</td>
</tr>
<tr>
<td>AREKF</td>
<td>1.9836</td>
</tr>
</tbody>
</table>

On the modeled orbital maneuver, the AREKF outperforms, once again, the other two algorithms. This is another example where the AREKF, in a RMSE-only based analysis, outperforms both the REKF and the EKF.
Conclusions and Recommendations
Final remarks of this thesis and recommendations for future work are presented in this chapter.

5.1 Conclusions

The work developed in this thesis concerned the study, implementation and analysis of three algorithms for state estimation of a spacecraft with an onboard GNSS receiver: EKF, REKF and AREKF. The main conclusions that can be drawn from this thesis are the following:

- A complete research on filter formulation was made. A theoretical analysis of the three algorithms and its stability conditions was performed. The characteristics, advantages and disadvantages were theoretically predicted and justified.

- Space environment modeling was performed. A perturbations model (SGP4) was implemented and an extensive effort was made on noise simulation. A method with a dynamic noise model was implemented.

- Two different scenarios of non-modeled accelerations were modeled and a scenario with measurement outliers were considered. This allowed to test the filters in several real space situations.

- Concerning the simulation results, all filters converged on the three environments. A fine tuning was performed for parameter $q_v$ for the EKF and an extensive analysis was made on both $\gamma$ and $\alpha$ parameters, for the REKF and AREKF, respectively.

- The REKF RMSE tends to the EKF’s monotonically crescent or decrescent, depending on the non-modeled accelerations and the tune of $Q$ and $R$.

- The REKF represents a reduction on the RMSE compared to the EKF only if non-modeled accelerations are relevant. Table 4.3 illustrates an example where it is not advantageous to use an REKF. However, since it is possible to revert a REKF to an EKF (enlarging $\gamma$), it is less time consuming implementing a REKF straight away when implementing both filters.

- The reduction of the RMSE of the AREKF varies between 0% and 63% less when compared to the EKF. This difference depends on the tuning of the $\alpha$ parameter and the intensity of the non-modeled acceleration. The AREKF outperforms in terms of RMSE the EKF and the REKF. The only possible case where this is not true is for a wrongfully tune of the $\alpha$ parameter by an order of magnitude.

- For the three environments considered the AREKF outperformed both competitors (assuming a proper $\alpha$ tuning): for a standard LEO orbit, for an orbit with outliers measurements and for spacecraft trajectory correction maneuvers.

- The EKF outperforms REKF that outperforms AREKF in terms of computational times. Being the algorithm with less calculations it is normal that the EKF is the fastest one. However, the algorithms revealed an overall short processing time. Depending on the application, this may be a very relevant factor (in space applications it is). No optimization to enhance computational times was made.
• As a final remark, the REKF and the AREKF revealed a short processing time, and completed the stated objectives of obtaining a simple to implement, robust, (adaptive in case of the AREKF) and low-power demanding algorithms.

5.2 Recommendations for Future Work

• Test the AREKF with real spacecraft data. None of the algorithms were tested with real data, but it is available to the public. Comparison between the algorithms could also be performed.

• An extensive study was made on the effect of the AREKF for a variable $\alpha$. The same study can be made - variation of RMSE versus the parameter - with $\gamma$. This was performed for the REKF, but not for the AREKF, which would be an asset.

• For different values of $q_v$ the $\alpha$ that minimized the RMSE was different. The higher the $q_v$ (less noticed non-modeled accelerations), the greater the $\alpha$, the higher was the RMSE. Intuitively, this means that the minimum of the function in Fig. 4.1 is moving upwards and to the right. However, to conclude in an exact formulation, more studies have to be carried out.

• A study of the tradeoff between performance of the filter and computational time has to be made for every situation, when deciding which algorithm to implement.

• An analysis on the influence of the number of visible satellites on different algorithms would be interesting. For both RMSE and a computational time based analysis.
Bibliography


A.1 Theorem 2.1

Proof of Theorem 2.1 (Xiong, 2009)

Choose a suitable Lyapunov function:

\[ V_k(\tilde{x}_k) = \tilde{x}_k^T P_k^{-1} \tilde{x}_k \quad (A.1) \]

By definition in Eq. 2.44 and assuming \( \nu_{min} = 1/p_{max} \) and \( \nu_{max} = 1/p_{min} \) the condition in Eq. 2.41 holds true and the function is bounded:

\[
\frac{1}{p_{max}} \| \tilde{x}_k \|^2 \leq V_k(\tilde{x}_k) \leq \frac{1}{p_{min}} \| \tilde{x}_k \|^2 \tag{A.2}
\]

To satisfy lemma 1 an upper bound on expression \( E[V_k(\tilde{x})|\tilde{x}_{k-1}] - V_{k-1}(\tilde{x}_{k-1}) \) in imposed by Eq. 2.41. Using the definition of prediction error and substituting Eq. 2.34 and Eq. 2.24 the prediction error of the REKF is:

\[
\tilde{x}_k = \beta_k F_k \tilde{x}_{k-1} - K_k H_k \beta_k F_k \tilde{x}_{k-1} + (I - K_k H_k) w_k - K_k v_k \tag{A.3}
\]

From Eq. A.1 and Eq. A.3 the conditional expectation can be written as:

\[
E[V_k(\tilde{x})|\tilde{x}_{k-1}] = E \left[ (\beta_k F_k \tilde{x}_{k-1})^T P_k^{-1} (\beta_k F_k \tilde{x}_{k-1}) - (K_k H_k \beta_k F_k \tilde{x}_{k-1})^T P_k^{-1} (K_k H_k \beta_k F_k \tilde{x}_{k-1}) \right] +
\]

\[
\left( \beta_k F_k \tilde{x}_{k-1} \right)^T P_k^{-1} (K_k H_k \beta_k F_k \tilde{x}_{k-1}) - (K_k H_k \beta_k F_k \tilde{x}_{k-1})^T P_k^{-1} (K_k H_k \beta_k F_k \tilde{x}_{k-1}) +
\]

\[
w_k^T (I - K_k H_k) P_k^{-1} (I - K_k H_k) w_k + v_k^T K_k^T P_k K_k v_k | \tilde{x}_{k-1} \tag{A.4}
\]

Through Eq. 2.37 and Eq. 2.36 it is possible to rewrite the Kalman Gain as \( K_k = P_k H_k^T R_k^{-1} \) and A.4 simplifies to:

\[
E[V_k(\tilde{x})|\tilde{x}_{k-1}] = \tilde{x}_{k-1}^T F_k^T \beta_k \left[ \Sigma_{k|k-1} - H_k^T (R_k^{-1} - R_k^{-1} H_k P_k H_k^T R_k^{-1}) H_k \right] \beta_k F_k \tilde{x}_{k-1} + \mu_k \tag{A.5}
\]

where \( \mu_k \) stands for

\[
\mu_k = E \left[ w_k^T (I - K_k H_k) P_k^{-1} (I - K_k H_k) w_k + v_k^T K_k^T P_k K_k v_k \right] \tag{A.6}
\]

From the first condition of Theorem 2.1 (Eq. 2.45) it is possible to derive:

\[
\Sigma_{k|k-1} \geq \beta_k F_k P_{k-1} F_k^T \beta_k + Q_k \geq \beta_k F_k P_{k-1} F_k^T \beta_k \tag{A.7}
\]

And it is also provable that:

\[
R_k^{-1} - R_k^{-1} H_k P_k H_k^T R_k^{-1} = (H_k \Sigma_{k|k-1} H_k^T + R_k)^{-1} \tag{A.8}
\]

Through the last two Eqs A.7 and A.8 and simplifying, we have:

\[
E[V_k(\tilde{x}_k)|\tilde{x}_{k-1}] - V_{k-1}(\tilde{x}_{k-1}) \leq \mu_k - \lambda_k V_{k-1}(\tilde{x}_{k-1}) \tag{A.9}
\]

where the parameter \( \lambda_k \) is:

\[
\lambda_k = \frac{1}{V_{k-1}(\tilde{x}_{k-1})} \left[ \tilde{x}_{k-1}^T F_k^T \beta_k H_k^T (H_k \Sigma_{k|k-1} H_k^T + R_k)^{-1} H_k \beta_k F_k \tilde{x}_{k-1} \right] \tag{A.10}
\]

The point of the proof is to bound the domain of the parameters \( \lambda_k \) and \( \mu_k \). For \( \mu_k \), it is legitimate to apply the trace operator to both sides and use Eq. 2.43 and Eq. 2.44 and determine the domain of the parameter:
\[ \mu_k \leq \text{tr}[(P_k^{-1} + H_k^T R_k^{-1} H_k P_k H_k^T R_k^{-1} H_k)Q_k] + \text{tr}(R_k^{-1} H_k P_k H_k^T) \]
\[ \leq (p_{\text{min}}^{-1} + h_{\text{max}}^2 r_{\text{min}}^{-2} p_{\text{max}})q_{\text{max}} l + h_{\text{max}}^2 p_{\text{max}} r_{\text{max}}^{-1} m \equiv \mu_{\text{max}} > 0 \quad (A.11) \]

Using the matrix inversion lemma to Eq. A.9

\[ V_k-1(\tilde{x}_{k-1}) - \lambda_k V_k-1(\tilde{x}_{k-1}) = \tilde{x}_{k-1}^T \left[ P_k^{-1} - F_k \beta_k H_k^T (H_k \Sigma_{k|k-1} H_k^T + R_k)^{-1} H_k \beta_k F_k \right] \tilde{x}_{k-1} \]
\[ > \tilde{x}_{k-1}^T \left[ P_k^{-1} + P_k^{-1} F_k^T \beta_k H_k^T (H_k Q_k H_k^T + R_k)^{-1} H_k \beta_k F_k P_k^{-1} \right] \tilde{x}_{k-1} > 0 \quad (A.12) \]

which implies that the parameter \( \lambda_k \) is smaller than 1. Using again both Eq. 2.43 and Eq. 2.44 we can finally bound the parameter:

\[ \lambda_k \geq p_{\text{min}} (h_{\text{min}} \beta_{\text{min}} f_{\text{min}})^2 [p_{\text{max}} (h_{\text{max}} \beta_{\text{max}} f_{\text{max}})^2 + q_{\text{max}} h_{\text{max}}^2 + r_{\text{max}}]^{-1} \equiv \lambda_{\text{min}} > 0 \quad (A.13) \]

and finally we obtain the bound for both parameters, in which \( \mu_{\text{max}} > 0 \), \( 0 < \lambda_{\text{min}} < 1 \) where the inequality holds:

\[ E[V_k(\tilde{x}_k)|\tilde{x}_{k-1}] - V_k-1(\tilde{x}_{k-1}) \leq \mu_{\text{max}} - \lambda_{\text{min}} V_k-1(\tilde{x}_{k-1}) \quad (A.14) \]

and we are able to conclude that the estimation error is bounded in mean square.
A.2 Theorem 2.2

Proof of Theorem 2.2 (Xiong, 2009)

To prove Theorem 2.2 the process taken is similar to the previous theorem, exception made on the assumption of the error covariance matrix, that in this case follows Eq. 2.54. The real covariance matrix of the innovation follows Eq. 2.24 and Eq. 2.44:

\[
\bar{P}_{y,k} = E \left[ (y_k - H_k \tilde{x}_{k|k-1}) (y_k - H_k \tilde{x}_{k|k-1})^T | \tilde{x}_{k-1} \right] = H_k \bar{\Sigma}_{k|k-1} H_k^T + R_k \tag{A.15}
\]

From the previous expression for \( P_{y,k} \) Eq. 2.35 and Eq. A.15:

\[
P_{y,k} - \bar{P}_{y,k} = H_k (\Sigma_{k|k-1} - \bar{\Sigma}_{k|k-1}) H_k^T \tag{A.16}
\]

Since is assumed that the rank of the measurement matrix \( H_k \) is \( l \) and \( \alpha = 1 \), if the left side of Eq. A.16 is positive, the condition \( \Sigma_{k|k-1} > \bar{\Sigma}_{k|k-1} \) is always fulfilled. When the condition \( P_{y,k} > \bar{P}_{y,k} \) does not hold true, using Eq. 2.52 and Eq. 2.54 yields:

\[
\Sigma_{k|k-1} - \bar{\Sigma}_{k|k-1} = \lambda_k P_{k|k-1} - \bar{\Sigma}_{k|k-1} > 0 \tag{A.17}
\]

which shows that \( \Sigma_{k|k-1} > \bar{\Sigma}_{k|k-1} \) is always positive and the condition in Eq. 2.48 is satisfied. The rest of the proof follows the approach of Theorem 2.1.
A.3 Lemma 2.1

Proof of Lemma 2.1 (Xiong, 2009)

Considering $V(\xi_k)$ a stochastic process with $\xi_k$ a stochastic process exponentially bounded in mean square and assuming for all $\xi_t \in \mathbb{R}$, $v_{\min}, v_{\max}, \mu > 0$ and $0 < \lambda < 1$ two conditions are imposed:

$$v_{\min} \| \xi_k \|^2 \leq V(\xi_k) \leq v_{\max} \| \xi_k \|^2 \quad (A.18)$$

and

$$E\{V(\xi_k)|\xi_{k-1}\} - V(\xi_{k-1}) \leq \mu - \lambda V(\xi_{k-1}) \quad (A.19)$$

From condition in Eq. A.19

$$E\{V(\xi_k)|\xi_{k-1}\} \leq \mu + (1 - \lambda)V(\xi_{k-1}) \quad (A.20)$$

Using the property (Tzyh, 1976)

$$E\{V(\xi_k)|\xi_{k-2}\} = E\{V(\xi_k)|\xi_{k-1}, \xi_{k-2}\} \quad (A.21)$$

One gets

$$E\{V(\xi_k)|\xi_{k-2}\} \leq E\{\mu + (1 - \lambda)V(\xi_{k-1})|\xi_{k-2}\} = \mu + (1 - \lambda)E\{V(\xi_{k-1})|\xi_{k-2}\} \quad (A.22)$$

Using the second condition in Eq. A.19 again,

$$E\{V(\xi_k)|\xi_{k-2}\} \leq k + (1 - \lambda)[k - (1 - \lambda)V(\xi_{k-2})] = k + k(1 - \lambda) + (1 - \lambda)^2V(\xi_{k-2}) \quad (A.23)$$

Applying this formula recursively and using the first condition in Eq. A.18

$$E\{\| \xi_k \|^2 \} \leq \frac{v_{\max}}{v_{\min}} E\{\| \xi_0 \|^2 \}(1 - \lambda)^k + \frac{\mu}{v_{\min}} \sum_{i=1}^{k-1} (1 - \lambda)^i \quad (A.24)$$

Remark:

Using

$$\sum_{i=1}^{k-1} (1 - \lambda)^i \leq \sum_{i=1}^{\infty} (1 - \lambda)^i = \frac{1}{\lambda} \quad (A.25)$$

The inequality in Eq. A.24 can be rewritten as

$$E\{\| \xi_k \|^2 \} \leq \frac{v_{\max}}{v_{\min}} E\{\| \xi_0 \|^2 \}(1 - \lambda)^k + \frac{\mu}{v_{\min}} \frac{1}{\lambda} \quad (A.26)$$
Further Results

Contents

B.1 Innovations Results ......................................................... B-2
B.2 AREKF RMSE Complete Results ......................................... B-3
B.1 Innovations Results

As an example, it is shown a plot of the innovations throughout a period of a LEO of an EKF in Fig. B.1.

![Figure B.1: Innovations for an EKF run with $q_v = 2(m/s)^2$ for a LEO orbit.](image)

It is possible to see the convergence of the filter on the first iterations of the algorithm, where the innovations decrease. This happened similarly on the other three filters.
B.2 AREKF RMSE Complete Results

The values obtained for the AREKF with the variation of parameter $\alpha$ are presented in Table B.1 for parameters $\rho = 0.98$, and a LEO orbit of period $T = 5923 \text{ secs}$. 

Table B.1: AREKF RMSE complete results for $\gamma = 1.00005 \sqrt{\text{max}(\text{eig}(P))}$ and $q_v = 0.001 (m/s)^2$ for 20 runs

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>AREKF RMSE [m]</th>
<th>EKF RMSE [m]</th>
<th>REKF RMSE [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0001</td>
<td>13.8596 ± 0.0118</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>13.3263 ± 0.0120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>12.5141 ± 0.0103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>11.4209 ± 0.0095</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>10.3667 ± 0.0108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>9.5454 ± 0.0099</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>9.0226 ± 0.0081</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>8.7786 ± 0.0083</td>
<td>13.4132 ± 0.0106</td>
<td>10.4564 ± 0.0074</td>
</tr>
<tr>
<td>0.7</td>
<td>8.7564 ± 0.0086</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>8.8927 ± 0.0073</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>9.0222 ± 0.0075</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9.1265 ± 0.0080</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10.0637 ± 0.0104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10.3423 ± 0.0085</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10.6092 ± 0.0104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10.7301 ± 0.0077</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>10.7941 ± 0.0106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>10.869 ± 0.0085</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>